自己設計 APPLE 遊戲

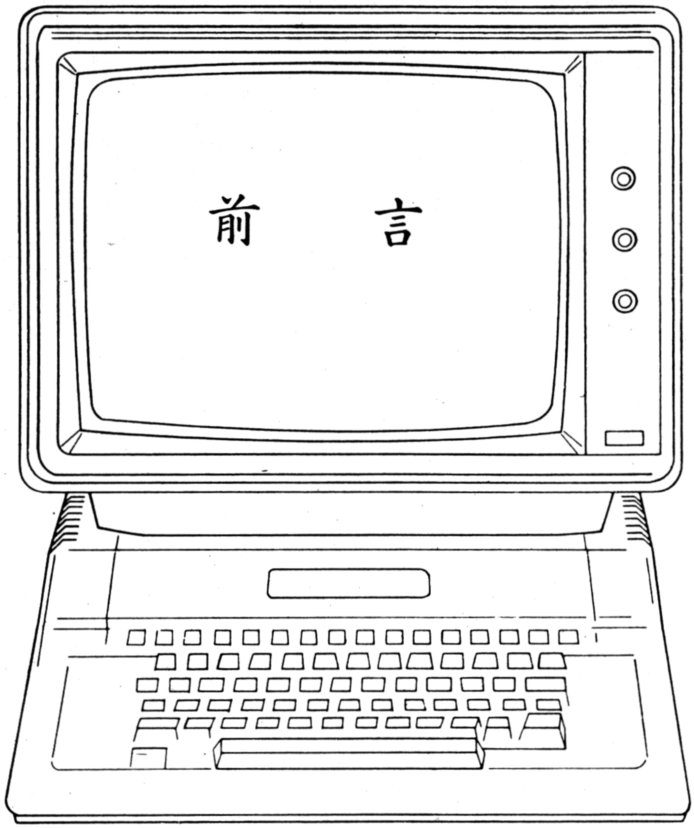


郭念台 譯

自己設計 APPLE 遊戲

郭念台 譯

儒林圖書公司 印行



首先，歡迎您加入Apple電算世界！Apple是一部設計極佳的個人電腦，它有極佳的繪圖能力，並且可接遊戲控制器，因此特別適合做電腦遊戲。本書含有四十種以上BASIC 遊戲和程式，它們都是特別爲Apple 設計的，可充分顯示Apple的優點。

這些遊戲是以48 k的Apple Ⅱ Plus爲準而寫的，而且也都在這種電腦測試過，但是因爲它們都是用標準 Applesoft Basic語言寫成的，因此應該可以在任何一部可使用該語言的Apple 上執行。這些遊戲大多需要接上Apple 遊

戲控制器，而且需要一部磁碟機來存取程式。注意，本書所示之遊戲圖片雖然都是以黑白印刷的，但是實際上這些遊戲程式都充分利用了 Apple 的彩色繪圖能力。

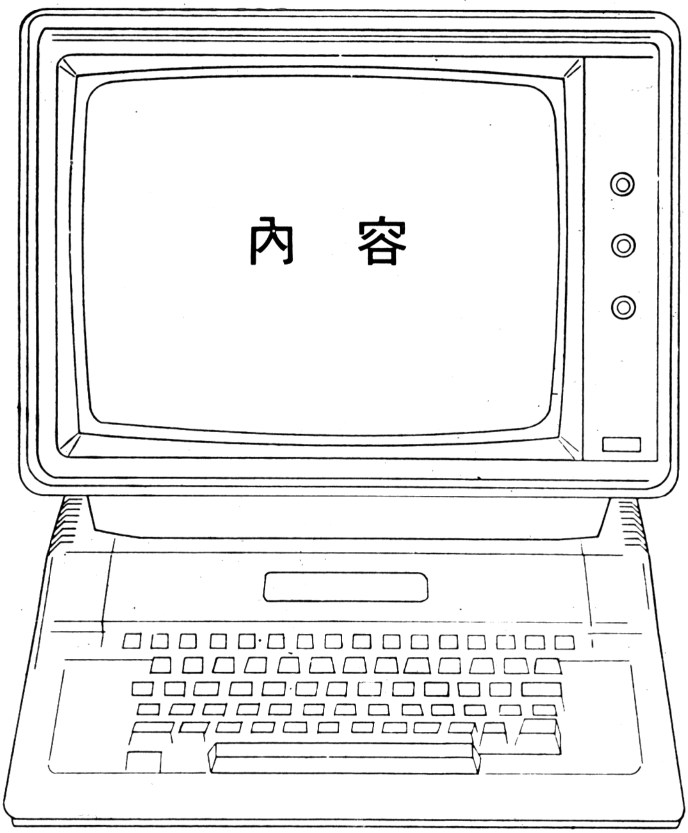
這些遊戲之中有些是老遊戲；例如"劊子手"（ Hangman ）和"井字遊戲"（ Tic-Tac-Toe ）等，有些是將市面上的程式改寫為 Apple 程式；例如"直昇機"（ Helicopter ）和"電位綫"（ Electric Field ）* 等。不過，大多數都是全新的設計。

所有遊戲程式都列在本書中，您可以把它們打入電腦， SAVE（存放）在磁碟片上，再叫出來玩。如果只是為了玩這些遊戲，您根本無需去懂 BASIC 與電腦，但是為了獲得更大樂趣，您可以學習如何充分利用您的 Apple 與使用手册自己設計 BASIC 遊戲。仔細硏讀書中各遊戲的說明，並試着玩玩那些遊戲，然後再試着設計自己的遊戲程式。

第一章為您介紹一系列您可以利用來設計遊戲程式的技巧，副程式，和邏輯架構（ algorithm ）。筆者假定您已經很熟悉 BASIC，希望您將第一章內的所有副程式打入電腦，執行它們，並試着去修改它們。仔細硏讀書中的遊戲程式，您將會發現這些副程式中有許多暗藏在那些遊戲程式內。

祝您成功！

* 譯者註：參閱"電位綫"那一節。



前　　言 ………………………………………………………… I

1.寫作遊戲程式之技巧 ……………………………………… 1

以模組爲單位寫作 ………………………………………… 1

讓遊戲者輪流玩 …………………………………………… 3

指定顏色給遊戲者 ………………………………………… 5

索取遊戲者之姓名，並指定一個遊戲控制器給他 ……… 7

多利用遊戲控制器…… 7
藉閃爍來吸引遊戲者…… 11
從鍵盤中斷程式執行之方法…… 13
字鍵代碼…… 15
利用鍵盤來控制動作…… 18
方格圖的畫法…… 19
在方格圖中選格子…… 20
在方格圖中移動…… 21
乒乓球拍…… 22
碼錶…… 23
記分…… 25
移動坦克…… 25
移用兩頁高解析度記憶頁的動態畫…… 27
高解析度圖形，形狀表，與商用繪圖軟體…… 28
圓，螺旋線與鏡面…… 29
多邊形，星形與旋形圖（ Squiral ）…… 34
平均值與插值法…… 37
爬行的字幕…… 39
亂數…… 40
整數 BASIC 中的亂數函數…… 40
Applesopt 的亂數函數…… 43
修截亂數…… 43

亂數可重複嗎？…… 44
零與壹…… 47
設定上下限…… 48
隨機時間延遲…… 48
東倒西歪（ Random walk ）…… 49
東倒西歪一加一或減一…… 50
不等步伐大小的東倒西歪…… 52
濾平（ Smoothing ）…… 53
不等頻率分佈的亂數…… 56
低解析度圖形的擲骰子遊戲…… 59
在遊戲程式中加註標題…… 61
徵求朋友的意見…… 63
一個好遊戲應該怎樣？…… 65

2.由選擇表開始動手…… 67

"Hello" 程式…… 67

3.動作遊戲…… 76

坦克大決戰（ Tank ）…… 76
衝刺（ Sprint ）…… 82
轟炸機（ Bomber ）…… 86
伺機而發（ Wait for It ）…… 90

賽馬（ Horse Race ）…… 93
打靶（ Target ）…… 96
直昇機（ Helicopter ）…… 99
緊迫盯人（ Tiddler ）…… 104
乒乓（ Ping Pong ）…… 106
太空突擊隊（ Space Zapper ）…… 109
美式骰戲 1（ Craps ）…… 113
美式骰戲 2（ Diddle ）…… 116
踩尾巴（ Cross Porpoises ）…… 119
舉重選手（ Weight Lifters ）…… 122

4.策略遊戲 …… 125

浪子回頭（ Zig-Zag ）…… 125
搭橋（ Bridge-It ）…… 129
股票市場（ stock Market ）…… 133
黃金大拍賣（ Gold Auction ）…… 137
井字棋（ Tic-Tac-Toe ）…… 142
翻面（ Turnover ）…… 147
吞食蛋糕（ Chomp ）…… 149

5.猜字遊戲 …… 153

劊子手（ Hangman ）…… 153

我的字是什麼？與 Wordlist 1 …… 157

拼字遊戲（ Anagrams ）…… 163

生日卡（ Birthday Card ）…… 168

貴姓大名（ What's Your Name ？）…… 170

訊息看板（ Any Messages？）…… 173

請勿亂動（ Do Not Adjust ）…… 174

6.繪圖副程式 …… 175

蝴蝶（ Butterfly ）…… 175

烟火（ Firework ）…… 177

高解低度與低解析度等電位圖 …… 179

李沙育圖（ Lissajou ）…… 182

蒲公英（ Dandelions ）…… 184

酒杯墊（ Doily ）…… 185

六角星圖（ Stars）…… 186

多邊形與旋形圖（ Polygons and Squirals ）…… 187

碼錶（ Stopwatch ）…… 189

7.單人遊戲 …… 193

記憶人（ Memory Man ）…… 193

風琴（ Organ ）…… 197

音樂（ Music）…… 202

迷宮（ Elithorn ）……205
爬坡（ Running on Empty ）……208
數字猜謎（ Guess a Number ）……210
尋寶（ Buried Treasure ）……214
換位遊戲（ Jumpers）……216
拔釘子（ Peggy）……219
智慧盤（ Fifteen）……223
開鎖（ Safebreak ）……227
耍把戲者（ Juggler ）……231

8.教育與科學程式……235

箭（ Arrow ）……235
東倒西歪（ Random Walk ）……239
擴散（ Scatter ）……243
巴斯卡三角形（ Pascal's Triangle）……245
魔術方塊（ Magic Squares ）……248
動畫描圖板（ Animated Sketchpad ）……253



以模組爲單位寫作

當您準備開始寫作一個遊戲程式時，最好先畫出它的流程圖，這樣可幫助您了解遊戲的邏輯架構，而且可將遊戲分解成若干較少的模組（ module ），然後您就可以以模組爲單位來設計。舉例而言，假定您想寫作"井字棋"（ Tic-Tac-Toe ）的電腦程式，這遊戲可能包含以下諸步驟：

1. 設定各個變數的初值

2. 畫出方格圖（或棋盤）
3. 輸入遊戲者要下的位置
4. 檢查要下的位置是否已有棋子
5. 在選定的位置上畫出代表遊戲者一方棋子之符號
6. 電腦根據策略決定要下的位置
7. 檢查該位置是否已有棋子
8. 畫出代表電腦這一方棋子之符號
9. 檢查看看是否有三個棋子在一條綫上；亦即，是否有人贏了
10. 回到程序3

上列步驟中1至9都是一個單獨的副程式，有些步驟看起來一樣，而且在一盤遊戲中要出現好幾次，因此可以合成一個副程式。無論是輪到遊戲者或電腦，每下一步之後您都必須檢查那一步是否符合規則；亦即，那一步是否已經有棋子。步驟4與7都是做同樣的事，因此很明顯地可以用一個副程式來做，讓它在每一回合中執行兩次。如果那一步符合規則，則您需要根據輪到那一方而呼叫畫叉或畫圓圈的繪圖副程式。無論是輪到遊戲者或電腦，每下一步之後您都需要呼叫前述繪圖副程式（步驟5或步驟8），再檢查看看是否有人贏了（步驟9）。現在，將這些步驟畫成流程圖，將各個副程式放在一個方框內，在需要做選擇的地方把問題放在一個菱形框內，並在指向下一步的箭頭旁邊標上"是"或"否"。

您現在可以思考如何着手寫作各個副程式了。首先，看看那些步驟需要遊戲者輸入情報給電腦，或者需要電腦回饋情報給他，儘量試着讓這些情報以使用者為導向（user-oriented）；亦即，儘量使說明與規則簡潔易懂，使遊戲者儘快了解玩法。記住一個原則：如果能夠在螢光幕上用圖形說明就千萬不要用文字說

明。此外，如果需要遊戲者輸入資料時千萬不要期望他是位專業打字員，要儘量使遊戲者所需的動作減至最少。如果您需要的是“是／否”（ YES ／ NO ）的答案就不要要求他打“ 0 ”來代表“是”，或以“ 1 ”來代表“否”。對電腦而言，答案同樣是一連串的 0 或／與 1 ，但是對人們而言，此種表示法稍嫌麻煩了一些，儘量讓他自然地回答“是”（ YES ）或“否”（ NO ）。不過，當您需要這種答案時，用“ Y ”來代表“ YES ”以及用“ N ”來代表“ No ”不是更簡單嗎！爲什麼要讓他打三個字呢？

關於遊戲情報的顯示，利用圖形顯示井字是一件很自然的事，但是倒底要使用高解析度圖形呢，還是低解析度圖形呢？對井字棋，這不是困難的問題—您只需要九個方格，每個格子能夠容納一個叉或一個圓圈即可。低解析度圖形就夠了，而且它會給遊戲者更堅實，更有色彩感。此外，低解析度圖形比高解析度圖形更適合步驟 4 的執行；步驟 4 的目的是檢查某個格子內是否已有棋子，利用低解析度的 SCRN 功能可輕易地完成這件事（SCRN 的功用是檢查某已知方格內的顏色）。高解析度圖形法中沒有 SCRN 功能，這使得人們傾向於使用低解析度圖形來顯示這個遊戲。用來畫方格與棋子符號的繪圖副程式並不困難，我們稍後會說明。

步驟 9 是一個檢查是否有三個棋子（圓圈或叉）在一條綫上的副程式。寫作遊戲程式的基本觀念就是先把它分成若干較小的模組，然後分別寫作這些副程式，再把它們組成一個完整的程式。接下去，我們要討論寫作這些模組的各種技巧。

讓遊戲者輪流玩

在所有兩人遊戲中，遊戲者都是輪流參與的；在電腦上最好的方法是把主程

式交給一位遊戲者，然後再交給另一位遊戲者。要執行這件事，只要定義一個變數P（代表Player），讓它在0與1之間交互變化，並決定輪到誰即可；例如，PDL（P）就可根據輪到誰去讀取0號控制器或1號控制器的數值。下面這段程式即可完成上述手法：

```
100 P = 0
110 P = 1 - P
120 PRINT P, PDL(P): GOTO 110
```

若想令P在1與2之間交互變化，可使用

```
100 P = 1
110 P = 3 - P
120 PRINT P, PDL(P - 1): GOTO 110
```

下面是較通用的程式，它適用於包括負數在內的任何一對數目：

```
 90 REM ALTERNATION
100 INPUT "TYPE IN TWO NUMBERS";A;B
110 PRINT "THIS PROGRAM WILL ALTERNATE
BETWEEN ";A;" AND ";B
120 P = A
130 P = (A + B) - P
140 PRINT P;" ";
150 GOTO 130
```

您可以把每個遊戲者的次數，積分等資料分別存放在陣列（array）中，您可以用下法宣告陣列：

```
10 DIM ASSETS (2),CAPITAL (2),TIME (2)
```

雖然您不一定要宣告它們，因為 Applesoft 會自動執行

```
DIM ASSETS (10)
```

只要您在程式中使用了它。如果一位遊戲者贏了10點，而他的對手輸了8點，您可以利用

```
145 ASSETS (P) = ASSETS (P) + 10
148 ASSETS (1 - P) = ASSETS (1 - P) - 8
```

來記錄他們的成績。（您不能用 SCORE 作爲變數名字，因爲 scORe 有保留字 OR 在裏面。）

指定顏色給遊戲者

在低解析度圖形遊戲中最好爲每位遊戲者分派一個顏色。深藍色（COLOR = 2）與黃色（COLOR = 13）是很合適的顏色，因爲它們在彩色電視螢光幕上看起來很醒目，而且即使在黑白電視螢光幕上仍然可以輕易地分辨出來，因爲深藍色看起來比黃色要暗。

您可以直接寫

```
10 IF P = 0 THEN COLOR = 2
20 IF P = 1 THEN COLOR = 13
```

不過，較佳的作法是

```
10 COLOR = 2 + 11 * P
```

在兩種顏色 C1 與 C2 之間交互變化的通式如下：

```
10 COLOR = C1 + (C2 - C1) * P
```

假定您決定令 P 在 1 與 2 之間交互變化，不是 0 與 1，則答案應爲：

```
10 COLOR = 2 + 11 * (P - 1)
```

，而其通式則爲

```
10 COLOR = C1 + (C2 - C1) * (P - 1)
```

若要通知遊戲者“輪到您了”，您可以寫

```
50 PRINT "YOUR TURN, PLAYER ";P
```

或者，您可遂行稱呼遊戲者的顏色，最簡單的方法是

```
10 IF P = 0 THEN PRINT "BLUE"
20 IF P = 1 THEN PRINT "YELLOW"
```

不過，較佳的解法是：

```
10 C$ = "BLUE  YELLOW"
20 FOR P = 0 TO 1
30 PRINT "YOUR TURN, "; MID$(C$,(1 + 6 * P),6)
40 NEXT P
```

C$ 包含兩個顏色名稱分字串（substring） 所組成的一個字串。因爲它們必須具有相等長度，因此在BLUE 後面補上兩個空格，使它們都成爲兩個 6 一字元的分字串。第一個分字串是" BLUE "，由C$ 的第一個字母開始，第二個分字串是" YELLOW"， 由C$ 的第七個字母開始。這兩個分字串由 30 那一列 PRINT （印）出來。記住

```
MID$(C$, A, B)
```

選擇的是由C$ 的第A個字元（ character ）開始而長度爲B 個字元的分字串。如果 P = 0 ，則 30 那一列中的（ 1 ＋ 6 * P ）求出之值爲 1 ，而如果 P = 1 ，則其值爲 7 。因此，程式執行的結果如下：

```
RUN
YOUR TURN, BLUE
YOUR TURN, YELLOW
```

C$ 包含兩個N一字元分字串的通式爲：

```
MID$(C$(1 + N * P),N)
```

例如，假定 0 號遊戲者被指派的顏色爲藍色（ BLUE ），而 1 號遊戲者爲紅色（ RED ），則C$ 包含的是兩個 4 一字元的分字串，因此 N = 4 ：

```
10 C$ = "BLUERED "
20 FOR P = 0 TO 1
30 PRINT "YOUR TURN, ";MID$(C$,(1 + 4 * P),4)
40 NEXT P
```

其結果爲

```
RUN

YOUR TURN, BLUE
YOUR TURN, RED
```

注意，C$的最後一個字元（在RED後面）是一個空格。

索取遊戲者之姓名，並指定一個遊戲控制器給他

遊戲者多半喜歡看到自己的姓名出現在螢光幕上，這樣可以提高他的興趣。下面是一個可直接稱呼遊戲者名字，並確保他使用正確控制器的副程式。

```
10 PL$ = "FIRST SECOND"
20 TEXT:HOME:VTAB 10
30 FOR J = 1 TO 2
40 PRINT MID$(PL$, 1+6*(J-1), 6); "PLAYER, PLEASE
TYPE YOUR NAME & HIT RETURN"
50 INPUT NAME$(J)
60 PRINT: PRINT "THANK YOU"
70 NEXT J
80 PRINT "EACH PLAYER NEEDS A PADDLE FOR THIS
GAME.  PLEASE TURN THE KNOB ON ONE PADDLE NOW."
90 FOR J = 0 TO 1
100 PAD(J) = PDL(J)
110 IF ABS (PDL(J) - PAD(J)) > 1 THEN Q = J: GOTO
140
120 NEXT J
130 GOTO 90
140 PRINT "THANK YOU.  THAT IS PADDLE # "; Q; ".
GIVE IT TO "; NAME$(Q + 1)
150 PRINT "(GIVE THE OTHER PADDLE TO ";
NAME$(2-Q); ")."
```

現在，您可以在程式中任何地方以NAME$(1)或 NAME$ (2)來稱呼遊戲者了。您以後會發現"太空突擊隊"（ Space Zapper ）中就使用了這個副程式。

多利用遊戲控制器

您可以多利用遊戲控制器來控制遊戲的進行，少利用鍵盤。每個控制器是一只 150 K歐姆的可變電阻器，Apple 利用 PDL（0）與 PDL（1）來讀取電阻值，其程式如下：

```
90 REM PRINT PADDLE SETTINGS---
100 TEXT: HOME
110 VTAB 10
120 PRINT "PADDLE # 0","PADDLE # 1"
130 VTAB 12
140 PRINT PDL(0);"   "
150 FOR T = 0 TO 25: NEXT T
160 PRINT PDL(1);"   "
170 GOTO 130
```

兩個控制器的數值會並排出現在螢光幕的第12列上。程式中140與160那兩列內的空格〃　〃用來避免當您旋轉控制器使讀數改變而前後讀數之位數不同時將不需要的數字仍留在螢光幕上。如果沒有這兩列，當控制器的數值由251變為9時您將會看到出現在螢光幕上的是〃951〃；9重疊在前面251的第一個數字上。Applesoft 使用手册中建議在讀取控制器讀數時在相鄰的兩次動作中最好加入一小段時間延遲，否則會發生錯誤，這也就是程式中150那一列的目的。

有些人可能覺得把這些數值以高解析度圖形畫在螢光幕上比顯示一堆數字會更令人覺得有興趣，其程式如下：

```
100 TEXT: HOME: HGR2
110 X = PDL (0)
120 FOR HC = 3 TO 0 STEP -3
130 HCOLOR = HC
140 HPLOT X, 50 TO X, 150
150 NEXT HC
160 GOTO 110
```

這段程式可在螢光幕上畫出一條對應於0號控制器位置的白色（HCOLOR = 3）垂直綫，然後在畫另一個位置之前把它擦掉。

下面是二維空間的程式；令PDL(0)控制一個高解析度點的水平位置，而令PDL(1)控制該點的垂直位置。

```
90 REM PADDLE ETCH-A-SKETCH-----
100 TEXT: HOME:HGR2: HCOLOR = 3
110 X = PDL (0) * 279/255
120 Y = PDL (1) * 191/255
```

```
130 HPLOT X, Y
140 GOTO 110
```

現在，讓我們再加入一個〞導向飛彈〞，您只要把下列幾列加入程式中：

```
110 MX = 250 * RND (1): MY = 190 * RND (2):HPLOT
MX, MY: REM INITIAL POSITION OF MISSILE
150 MX = MX + SGN (X - MX): REM HOMING ALGORITHM
160 MY = MY + SGN (Y - MY)
170 HPLOT MX, MY
```

再執行（RUN）程式。前面的〞ETCH - A - SKETCH〞部分仍會產生一個受控制器控制的點，但是多了一個由任意位置（MX，MY）發射，但不論您如何旋轉控制器去逃避它都會鍥而不捨地追着您的點。這是怎麼做的呢？在程式中，150 那一列先測知位於MX 的飛彈倒底在位於X的目標的左方或右方。如果飛彈在左方，則（X－MX）爲正，因此SGN（X－MX）的數值爲1，使MX加1，令飛彈朝向目標飛行。如果飛彈在目標的右方，則SGN（X－MX）將等於－1，使飛彈移向左方—仍然朝向目標飛行；同理可分析 160 那一列中垂直方向的座標MY，Y。（SGN（A）表示〞A的符號〞。）

應用在實際飛彈上的原理也是一樣的：先偵測出目標至飛彈的〞誤差信號〞（距離）大小及方向，然後設法降低它。這種誤差降低法就是一般人所謂的〞負回授〞（negative feedback）；當誤差信號爲零時就表示飛彈正中目標。現在，看看您是否能將這種〞歸向〞（homing）原理應用到您自己設計的遊戲中。

另外有一種遊戲是去偵測那一位遊戲者轉動控制器比較快。這種方法應用的範圍也很廣；例如，在〞賽馬〞（Horse Race）和〞舉重選手〞（Weight Lifter）遊戲中，遊戲者必須儘快地來回旋轉控制器，那一位旋轉得快他就是（暫時的）贏家，其程式如下：

```
NEW

100 REM WHICH PADDLE MOVES MORE?---
110 TEXT: HOME
120 FOR J = 0 TO 1
130 PAD (J) = PDL(J)
140 FOR T= 0 TO 25: NEXT J
150 TRY (J) = ABS (PDL (J) - PAD (J))
160 IF TRY (J) > TRY (1-J) THEN VTAB 10: HTAB 5:
PRINT "PADDLE # ";J;" MOVES MORE"
170 NEXT J
180 GOTO 120
```

下面是另外一種應用控制器的方法：在〝百米衝刺〞(Sprint) 遊戲中，電腦產生一個來回游動的假想隨機指標，遊戲者則要儘快地來回旋轉各人的控制器，就好像瞎子在草堆中尋找一根針一樣，設法使各人的控制器讀數吻合那隱匿不見的指標位置，最靠近它的人得分，其程式如下：

```
100 REM PADDLES LOOKING FOR A POINTER---
110 TEXT;HOME
120 R = INT (255 * RND (1));REM RANDOM POINTER
130 A = ABS (R - PDL (0));REM MISALIGNMENT BETWEEN
PADDLE AND HIDDEN POINTER
140 B = ABS (R - PDL (1))

150 P = (A>B);REM TRUE = 1, 0 = FALSE
160 VTAB 10: PRINT "PLAYER ";P;" IS NEARER"
170 R = R + 10 * 20 * (RND(2) > 0.5): REM ADD OR
SUBTRACT 10 TO RANDOM POINTER
180 IF R < 10 THEN R = 40

190 IF R > 250 THEN R = 220
200 GOSUB 250: REM PRINTOUT
210 GOTO 130
250 REM PRINT POSITIONS OF PADDLES & POINTER---
260 FOR L = 1 TO 4: PRINT "
";NEXT L: REM CLEAR LINES 1 TO 4
270 H(J) = PDL (J) * 40 / 255
280 VTAB J + 1: HTAB H (J) + 1: PRINT J;REM PRINT
PADDLE POSITIONS
290 NEXT J
300 HR = R * 40 / 255
310 VTAB 4: HTAB HR: PRINT R: REM PRINT POINTER
POSITION
320 RETURN
```

看不見的指標 R 在 120 那一列中被設定爲一個介於 0 與 255 之間的隨機値，並在 170 那一列中隨機上下游動，而遊戲者則設法旋轉各人的控制器去找它；遊

戲者的誤差分別爲130與140那兩列中的ABS（R—PDL(0)）與ABS（R— PDL（1））。150與160那兩列決定那一個人的誤差較小，程式的其餘部份則給予較近的遊戲者（P）適當的點數。250至320那幾列內的副程式會將過程顯示在螢光幕上：兩個控制器的位置由280那一列中的HTAB指令顯示，而指標的位置則由310那一列顯示。

您可以改寫上列控制器副程式使它們適合您的需求。控制器的阻值只要介於0至150K歐姆之間變化，Apple便能夠接受，仍然會認爲是一只控制器；因此，如果您把合適的電阻性裝置，諸如光電池、熱敏電阻等，插入遊戲埠（ game port），上述的高解析度控制器描圖程式便可將您的Apple轉變爲一部光度計或溫度計。您甚至可以利用Apple經由遊戲埠去量度皮膚的阻值，把Apple轉變爲自製的測謊器。（參閱1982年12月出版之Creative Computing。）

藉閃爍來吸引遊戲者

在某些遊戲中，您可能會爲了想吸引遊戲者的注意力而使一塊彩色小方塊閃爍，您也可能會希望遊戲者在這個時候敲一下鍵去改變那小方塊的顏色。下面就是可達到這目的的程式：

```
80 GR
90 X = 20: Y = 20
100  FLICKER = 1
110 HUE = 9
120 COLOR = HUE * FLICKER
130 HLIN 0,39 AT 20
140 FLICKER = 1 - FLICKER
150 KEY = PEEK (-16384)
160 IF KEY > 127 THEN POKE -16368,0: HUE =  KEY
200 GOTO 120
```

140那一列使FLICKER在0與1之間交互變化；每經過迴圈（ loop ）一

次即變化一次。這也將使120那一列中的Color在HUE與零之間交互變化；亦即讓小方塊閃爍。HUE 在進入迴圈之前在110那一列內被隨意設定爲9（橙色）。150那一列偵測鍵盤，看看是否有鍵被按下，如果有，160那一列便以PEEK-16368，。重置鍵盤的觸發綫（Strobe），然後將HUE改爲KEY的數值，這使得控制跳出迴圈，讓該小方塊保持穩定（不閃爍）的橙色（COLOR＝9）。KEY的數值（每個鍵的數碼）等於該鍵字元的ASCII碼（參閱"從鍵盤中斷程式的執行"一節），這些數目介於128至255之間，但是當它被用作COLOR（顏色）的數值時會被自動轉換爲Mod 16；亦即，18變成2，19變成3，…，以此類推。因此，在160那一列隨便敲一個鍵便會改變HUE，並將畫圖的COLOR改爲0（黑色）與15（白色）之間的數目。

下面是另外一種讓一個小方塊閃爍的方法：

```
5  GR
10 FOR C = 0 TO 15 STEP 15
20   COLOR = C
30   PLOT 20, 20
40   FOR J = 1 TO 20
50     X = PEEK (-16384)
60     IF X > 127 THEN GOTO 100
70   NEXT J
80 NEXT C
90 GOTO 10
100 POKE -16368, 0: REM CLEAR KEYBOARD
110 IF C = 0 THEN COLOR = 15: PLOT 10, 10
120 COLOR = 9
130 PLOT 10, 10
```

程式中10至80這幾列所構成的迴圈會使一個位於20，20的小方塊藉著交互改變顏色（COLOR 0（黑色）與15（白色））而閃爍。如果您省略40至70那幾列（J－迴圈），則閃爍的速度會變得太快。J－迴圈有兩項作用：作爲降低閃爍速度的延時迴圈，和等待從鍵盤的輸入。如果遊戲者敲了鍵，則60那一列內的X＞127條件將被滿足，使程式跳出迴圈到100那一列將鍵盤清除（clear）。110那一列檢查位於20，20的小方塊是否被無意中設定爲黑色（C=COLOR

＝0 ），如果是，則將它矯正爲白色，然後繼續做別的事；此處，120 至130 那幾列是在位置10 ，10 處畫一個橙色小方塊。

從鍵盤中斷程式執行之方法

有三種方法將資料經由鍵盤輸入正在執行中的程式：第一種方法是照下法使用INPUT 指令

```
100 INPUT "HOW LONG DO YOU WANT THE
LINE..."；L
```

程式便會停下來，一直到您輸入數目並按下RETURN（回頭）鍵才繼續下去。

您也可以利用GET指令來獲得資料，但是您無需按RETURN鍵。

```
100 PRINT "HOW LONG DO YOU WANT THE LINE TO
BE?"
110 GET L
```

同樣地，程式也會停下來，一直到您打入數目才繼續下去，不過，您無需按RETURN 鍵 ，只要您敲了鍵，程式就會立刻恢復執行；這種方法比較快速，對遊戲者也比較方便。GET與INPUT 不同之處是它不會把遊戲者回答之值顯示在螢光幕上，如果您需要它，您可以加入：

```
120 PRINT "LENGTH = "；L
```

萬一遊戲者不愼敲錯了鍵，譬如把數目敲成字母，程式會顯示出 SYNTAX ERROR ，並停下來。您可以用下面程式來避免此種情況發生：

```
100 PRINT "HOW LONG DO YOU WANT THE LINE TO
BE?"
110 GET L$：L = VAL(L$)
120 IF L<1 OR L>9 THEN GOTO 100
130 PRINT "L = "；L
```

現在，如果您敲的是字母鍵，L 會等於零，令程式回到100 那一列。

利用 GET 指令只能獲取一位數目，如果您需要的是二位數，其程式如下：

```
100 PRINT "HOW LONG DO YOU WANT THE LINE TO
BE?"
110 GET A,B
120 L = 10 * A + B
130 PRINT "L = ";L
```

您可能會覺得這樣做比使用 INPUT 麻煩多了，而且，如果 L＝5 ，您還必須打成 05 。

如果您希望從鍵盤輸入辭句，您可以使用 GET 指令：

```
100 DIM W$(4)
110 PRINT "TYPE IN A FOUR-LETTER WORD"
120 FOR J= 1 TO 4
130 GET W$(J)
140 NEXT J
```

您也可以使用 INPUT指令：

```
100 PRINT "TYPE IN A WORD  ";
110 INPUT W$
```

您將會注意到 GET 指令給您的是一個字串陣列 W $(4)，而 INPUT 指令給您的是一個字串 W$ 。

INPUT和 GET 指令都會使程式停下來，一直到獲得輸入才繼續下去。不過，有時候您可能想在程式眞正執行時去中斷它，此時您可以利用 PEEK(−16384) 指令：

```
100 PRINT TAB (20*RND(1));"PROGRAM RUNNING"
110 KEY = PEEK (-16384)
120 IF KEY > 127 THEN PRINT:PRINT "YOU
PRESSED KEY ";CHR$(KEY-128):PRINT
130 POKE -16368,0: REM CLEAR KEYBOARD
140 GOTO 100
```

現在，刪去 130 那一列，看看它有什麼影響。程式好像記得最後敲入的鍵，一直不斷地印出該鍵，這是我們不希望的現象，因此您必須記得在讀取一個鍵之

後要立刻將鍵盤重置（ reset ）。

字鍵代碼

在前一節中您可曾注意到我們使用過 CHR$（KEY—128）函數？在鍵盤上，每一個字鍵（A，B，C，…，1，2，3，…，*？；/， 等）都對應著一個數碼，此數碼剛好等於它的ASCII 碼加上128。如果您想看看這些數碼是什麼，可執行下面的小程式：

```
100 KEY = PEEK (-16384)
110 IF KEY > 127 THEN PRINT
CHR$(KEY-128),KEY,(KEY-128);POKE -16368,0
120 GOTO 100
```

當您執行這程式時您不會看到什麼，除非您敲了字鍵。每當您按下一個字鍵，螢光幕便會顯示出您敲入的字元，並在其後跟着兩個數目：一個是字鍵代碼，另一個是它的 ASCII 數碼。您將會注意到數目的ASCII 碼正好等於該數目加上48（0＝48，1＝49），而字母的ASCII 碼等於該字母在英文字母中的排行（A＝1，B＝2）加上64；因此A＝65，B＝66，…，Z＝90。這些字元的字鍵代碼又等於各個ASCII 碼加上128：對數目而言，它們等於數目本身加176（0＝176，1＝177）；對字母而言，它們等於各字母的排行加上192（A＝193，B＝194，…，Z＝218）。爲了方便對照起見，我們將各字元的ASCII 碼列在表1內。

表1. ASCII 碼

Dec=ASCII 十進位碼

Hex=ASCII 十六進位碼

Dec	Hex	鍵盤字元	Dec	Hex	鍵盤字元
0	00	ctrl @	48	30	0
1	01	ctrl A	49	31	1
2	02	ctrl B	50	32	2
3	03	ctrl C	51	33	3
4	04	ctrl D	52	34	4
5	05	ctrl E	53	35	5
6	06	ctrl F	54	36	6
7	07	ctrl G	55	37	7
8	08	ctrl H & ←	56	38	8
9	09	ctrl I	57	39	9
10	0A	ctrl J	58	3A	:
11	0B	ctrl K	59	3B	;
12	0C	ctrl L	60	3C	<
13	0D	ctrl M & CR	61	3D	=
14	0E	ctrl N	62	3E	>
15	0F	ctrl O	63	3F	?
16	10	ctrl P	64	40	@
17	11	ctrl Q	65	41	A
18	12	ctrl R	66	42	B
19	13	ctrl S	67	43	C
20	14	ctrl T	68	44	D
21	15	ctrl U & →	69	45	E
22	16	ctrl V	70	46	F
23	17	ctrl W	71	47	G
24	18	ctrl X	72	48	H
25	19	ctrl Y	73	49	I
26	1A	ctrl Z	74	4A	J
27	1B	ESC	75	4B	K
28	1C	n/a	76	4C	L
29	1D	ctrl-shift-M	77	4D	M
30	1E	ctrl ^	78	4E	N
31	1F	n/a	79	4F	O
32	20	space	80	50	P
33	21	!	81	51	Q
34	22	"	82	52	R
35	23	#	83	53	S
36	24	$	84	54	T
37	25	%	85	55	U
38	26	&	86	56	V
39	27	'	87	57	W
40	28	(	88	58	X
41	29	)	89	59	Y
42	2A	*	90	5A	Z
43	2B	+	91	5B	n/a
44	2C	,	92	5C	n/a
45	2D	-	93	5D	(shift-M)
46	2E	.	94	5E	^
47	2F	/	95	5F	n/a

從 96 至 255 的 ASCII 碼即爲用來產生 Apple 螢光幕字元的字鍵代碼；先重複右半邊的字元一次，再重複整個系列。雖然 CHR$(65) 與 CHR$(193) 都代表 A，但是對 Applesoft 而言，它們並非同一個字；而且，對印表機而言，它們也不是同一個字。

利用鍵盤來控制動作

您可以規劃鍵盤使您能夠隨心所欲地控制遊戲的進行。下面是一個可在高解析度圖形螢光幕上移動一個點的程式；在本程式中，E，X，S，D鍵可分別將一個點上、下、左、右移動。

```
80 REM KEYS MOVE A HIRES SPOT-----
90 ONERR GOTO 100
100 TEXT: HOME: HGR2: HCOLOR = 3
110 X = 140: Y = 96: HPLOT X, Y
120 W$ = "EXSD": REM DEFINE THE EFFECTIVE KEYS
130 REM WAIT FOR KEYPRESS-----
140 KEY = PEEK (-16384)
150 IF KEY > 127 THEN GOSUB 200
160 GOTO 140
200 REM READ KEYBOARD-----
210 POKE -16368, 0:REM CLEAR KEYBOARD
220 KEY = KEY - 128
230 FOR J = 1 TO 4
240 IF KEY = ASC (MID$ (W$,J,1)) THEN K = J: DX =
(K=4) - (K=3): DY = (K = 2) - (K = 1): GOSUB 300:
RETURN
250 NEXT J
300 REM REPLOT POINT-----
310 HCOLOR = 0
320 HPLOT X, Y
330 X = X + DX: Y = Y + DY
340 HCOLOR = 3
350 HPLOT X, Y
360 RETURN
```

130 至 220 那幾列是您已經熟悉的。120 那一列將E，X，S，D諸鍵置於字串W$中，再指定它們的任務。您也可以用其他任何四個鍵來取代它們；比方說U，D，L，R或者您熟悉的I，M，J，K游標移動鍵。如果您寫的是兩人操作的遊戲，您可以定義兩個字串，W$(1)=〃EXSD〃和W$(2)=〃IMJK〃，讓遊戲者肩靠肩並排坐在鍵盤前面分別控制屬於各人的那一組鍵。

下面是利用字鍵移動點的較直截了當，但也較笨拙的方法：

```
230 K$ = CHR$(KEY)
242 IF K$ = "S" THEN DX = -1
244 IF K$ = "D" THEN DX = +1
246 IF K$ = "E" THEN DY = -1
248 IF K$ = "X" THEN DY = +1
```

前一個程式中的 240 那一列比較漂亮，但是也比較難懂，因爲它有兩項改進：第一，它使用了布林邏輯（ Boolean logic ）。242 至 248 那幾列可精簡爲：

```
DX = (K$ = "D") - (K$ = "S")
DY = (K$ = "X") - (K$ = "E")
```

對布林運算式K$ = 〃D〃 而言，如果它爲眞（ true ）則取值爲1，如果爲僞（ false ）則取值爲0 ；同理可應用於K$= 〃S〃 ，K$= 〃X〃 ，或K$= 〃E〃 。所以，如果K$ = 〃D〃 ，則DX之值爲 ＋1（亦即，D鍵被按了），而如果K$ = 〃S〃 ，則 DX 爲－1 ；如果E鍵被按了，則DY 爲－1 ，而如果X鍵被按，則 DY 爲＋1 。

第二項改進爲比較被按的鍵與字串 W$(= 〃EXSD〃 ）。例如，X是 W$ 字串的第二個字元；換句話說，如果 X 鍵被按，則

```
KEY = ASC (MID$ (W$, 2, 1)
```

運算式 K$ = 〃X〃 可用更精簡的式子（ K = 2 ）來取代。將控制用字鍵置於字串中，然後放在程式前面的最大好處是隨時可以很容易地改變控制字鍵；比方說，如果要修改第二個例子中的 242 至 248 那幾列便需要修改很多地方，而且如果程式中還有別處有 K$ ，您還必須去把它找出來。但是，如果您把控制字鍵放在字串中，則當您要改變字鍵時只要修改 120 那一列即可。

方格圖的畫法

有許多棋戲，諸如井字遊戲、奧賽洛（ Othello ）等，都需要一幅有一些垂

直線與水平線所組成的方格圖（或棋板）。這項工作比較容易，其程式如下：

```
100 GR
110 COLOR=15
120 FOR I=0 TO 36 STEP 4
130    HLIN 0,36 AT I
140    VLIN 0,36 AT I
150 NEXT I
999 END
```

在方格圖中選格子

一旦您畫好方格圖或棋板後，您通常需要在板中選定某個格子，並且佔據它，就好像您在下西洋棋、井字遊戲時移動棋子一樣。您可以從鍵盤上敲入格子的座標，也可以利用控制器來選格子；後者比較方便。當您將控制器從 0 旋轉至255時螢光幕上有一個彩色小方塊會從左至右沿着上面一列移動，然後跳至第二列的左邊，掃過第二列，接着第三列，…，其餘的以此類推，就好像您在讀書一樣。

在100 那一列中遊戲者輸入他所希望的棋板大小；可從 3 × 3 選至 9 × 9 。程式的心臟在 120 到 150 那幾列，其作用是把控制器從 0 轉到 255 的讀數轉換成 X 與 Y 數值。如果 WIDE 設定爲 6 ，而 HIGH 爲 8 ，則 Y 會從 1 慢慢進到 8 。剛開始時，Y 停在 1 上，而 X 很快地從 1 進到 6 ；接下去，Y 變爲 2 ，X 又由 1 進到 6 ；然後 Y 變爲 3 ，…，以此類推。由 X 和 Y 決定的那個小方塊是在 170 那一列被畫在螢光幕的，裡面的 20 是隨意加上去的，其目的是要把畫面從左上角移到螢光幕的中央。160 那一列隨機決定顏色。

以下就是整個副程式

```
100 INPUT "HOW MANY SQUARES WIDE ";WIDE
105 INPUT "HOW MANY SQUARES HIGH ";HIGH
110 GR
120 DIV = 256/(WIDE * HIGH)
130 PADDLE = PDL(1)/DIV
140 X = WIDE * (PADDLE/WIDE -
INT(PADDLE/WIDE)): REM PADDLE MOD WIDE
150 Y = INT (PADDLE/WIDE)
160 COLOR=PDL(1)/8 + 1
170 PLOT X+20, Y+20
180 GOTO 130
```

在方格圖中移動

上面兩個副程式可合併爲一個副程式；事實上它早已爲許多程式所採用，例如尋寶遊戲(Buried Treasure)等。

```
5 TEXT: HOME: VTAB 10
10 INPUT "HOW MANY SQUARES WIDE "; WIDE
15 INPUT "HOW MANY SQUARE HIGH ";HIGH
20 REM DRAW GRID -----
30 GR: COLOR=15
40 FOR I = 0 TO 4 * WIDE  STEP 4
50 HLIN 0, 4 * HIGH AT I
60 NEXT I
70 FOR I = 0 TO 4 * HIGH STEP 4
80 VLIN 0, 4 * WIDE AT I
90 NEXT I
120 REM PADDLE # 1 DETERMINES POSITION OF
SQUARE -----
130 DIV = 256/(WIDE * HIGH)
140 PADDLE=PDL(1)/DIV
145 POLD=PDL(1)
150 X =  WIDE * (PADDLE / WIDE - INT
(PADDLE / WIDE)): REM PADDLE MOD WIDE
160 Y = INT (PADDLE / WIDE)
170 X = 4 * X + 2
180 Y = 4 * Y + 2
190 COLOR=9
200 PLOT X,Y
210 IF PEEK ( -16286) > 127 THEN GOSUB
300: REM READ PUSHBUTTON # 1 AND PAINT SQUARE
220 IF ABS(POLD-PDL(1))>1 THEN GOTO 250:
REM IF PADDLE MOVED THEN ERASE OLD POSITION
230 GOTO 220
240 REM ERASE OLD POSITION -----
250 COLOR = SCRN (X + 1, Y + 1)
260 PLOT X,Y
270 GOTO 140
300 REM PAINT SQUARE -----
310 COLOR = 9
320 FOR L = -1 TO +1
330 HLIN X - 1, X + 1 AT Y + L
340 SOUND = PEEK (-16336) + PEEK (-16336)
350 NEXT L
360 RETURN
```

5至 15 那幾列輸入所需的方格圖的高度與寬度，而由20至90那幾列把它畫出來。120至200 那幾列讀取1號控制器的數值，並將它轉換爲X和Y座標，再在170和180 那兩列中將其數值按棋板比例予以適當調整，然後在190和200那兩列中畫出來。當旋轉控制器時指標小方塊（ Pointer Square ） 沿著最上面一列由左向右移動，然後跳到第二列，第三列，…，好像在讀書一樣。220 那一列確保每當控制器移動時小方塊的前一個位置一定會在其新位置畫出來之前被250至270 那幾列清除掉。一般說來，人們要清除某個方塊都是在其上以COLOR＝0 再畫一次的，但是250 那一列却是把COLOR 設定爲SCRN(X＋1，Y＋1）；換句話說，它是先偵知指標小方塊所處方格的顏色，然後將小方塊也畫爲該顏色。如果遊戲者按了控制器，210 那一列會偵測出來，將控制器所處那一格塗上顏色，同時發出一聲短暫的蜂鳴聲。

如果您想將這個程式轉換成整數BASIC ，可將150和160 那兩列改爲更簡潔的形式：

```
>150 X = PADDLE MOD WIDE
>160 Y = PADDLE / WIDE
```

乒乓球拍

下面是一個在低解析度圖形中非常有用的副程式，它可在螢光幕上畫出一對可用控制器控制而上下移動的乒乓球拍，它可用於電視乒乓球遊戲中來拍打彈跳中的球。

```
10 REM PONGBAT-----
20 TEXT: HOME: GR
30 PAD (P) = INT (PDL (P)/7.5)
40 IF PAD (P) <1 THEN 70
50 REM BLACK LINE ABOVE AND BELOW PADDLE
60 COLOR = 0: VLIN 0, PAD (P) AT 39 * P
70 COLOR = 0: VLIN PAD (P) + 4, 39 AT 39 * P
80 REM PLOT BLUE OR YELLOW PADDLE
90 COLOR = 13 - 7 * P: VLIN PAD (P), PAD (P)
+ 4 AT 39 * P
100 P = 1 - P: REM CHANGE PLAYER
110 GOTO 30
```

90 那一列會在您希望的位置上畫出一條彩色的短垂直線來代表拍子，它位於PAD(P)與PAD(P)＋4之間，佔據4塊低解析度小方塊的高度；當然，拍子的位置是由30那一列中的控制器數値所決定的。PAD(P)的最大數值爲255／7.5＝34。0號控制器在螢光幕左邊畫的是黃色（COLOR＝13）拍子，1號控制器在螢光幕右邊畫的是藍色（COLOR＝6）拍子，而60與70那兩列則在拍子的上下各畫一條黑色的垂直線，這樣可在您移動控制器時自動擦去拍子的舊位置。

變數P（代表遊戲者）的數值在100那一列中交互變化於0與1之間，它快速輪流控制0號與1號控制器，使每個遊戲者覺得螢光幕上拍子的位置變化好像是連續的。

您可以在乒乓球遊戲（Ping Pong）中找到這個副程式和一個下面要說明的〝跳動的球〞（Bouncing Ball）副程式。

碼錶

假設您有一個需要計時的遊戲，您可能會想要在螢光幕上顯示一個數字鐘。下面就是一個在TEXT模式中很簡易的做法：

```
10 S = - 16336
20 N = 0
30 HOME
40 TICK = PEEK (S) + PEEK (S) - PEEK (S)
50 VTAB 21: HTAB 20: PRINT N
60 FOR PAUSE = 1 TO 700: NEXT PAUSE
70 N = N + 1
80 GOTO 30
```

這個副程式會將耗用的秒數顯示在螢光幕底部的中央（留下螢光幕的其餘空間供繪圖或做其他事情用）。N是耗用的秒數，在50那一列中印出來，40那一列使時鐘嘀嗒嘀嗒地響，而60那一列爲計時迴圈（timing loop），可提供相當標準的1秒計數。（您可以利用碼錶試試看將它改爲一分鐘或兩分鐘。）70那一列將N加1，而80那一列則跳回開頭處重新來過。

這個程式雖然很不錯，但是第六章中的碼錶（Stopwatch）程式因爲可以用低解析度圖形顯示出數字而顯得更令人喜愛。它是利用七段線條來組成一個數目字，就好像數字鐘上的七段顯示板（seven segment display）一樣，每一段是用簡單的HLIN或VLIN來形成副程式，我們將到第六章再詳細討論。

您想把您的Apple變成一個包括時、分和秒的數字鐘嗎？（它很可能是世界上最貴的電子鐘。）很簡單：

```
50 REM A DIGITAL CLOCK SHOWING HOURS,
60 REM MINUTES, AND SECONDS,
70 REM IN A 12 HOUR SEQUENCE.
100 TEXT: HOME
110 VTAB 10
120 HTAB 18
130 PRINT H;":";M;":";S;"  "
140 FOR T= 1 TO 700: NEXT T
150 S = S + 1
160 IF S > 59 THEN M = M + 1: S = 0
170 IF M > 59 THEN H = H + 1: M = 0
180 IF H > 12 THEN H = 1: M = 0: S = 0
190 GOTO 110
```

在這個很簡單，但有效的程式中，H，M和S各代表時，分和秒。110 至 130 那幾列將時間顯示在螢光幕上。注意時間顯示右邊的那兩個空白，這兩個空白可將前面顯示時間所留下來的殘餘數字除去；如果您省略了這兩個空白，則當時間由六位數變為五位數時，譬如由 12：59：59 變為 1：00：00 時，您將會看到末尾有一個額外的 9，變成——1：00：009。

140 那一列包含計時迴圈，您可以利用碼錶來核對，調整該計時數值。160 那一列在秒數到達 59 時將分進位，並將秒數重置為零，170 和 180 那兩列則主管分和時的進位與重置。您若想將它改為一個 24 小時的時鐘只要將 180 那一列改為：

```
180 IF H>24 THEN H = 1: M = 0: S = 0
```

記分

當遊戲正在進行時人們常常希望把遊戲者的分數顯示在螢光幕上。您可以只利用簡單的PRINT 敍述便可在畫面底部保留作文字說明的 4 列裡顯示分數；這種方法雖然可行，但是好像太平淡無奇了。有一個比較好的構想是利用低解析度圖形來顯示分數：您只要利用上面的〝碼錶〞副程式，改變R和C（列和行）的數值重新調配數字的位置，拋棄計時迴圈，並令變數TENS和UNITS 等於遊戲者的分數就可以了。

移動坦克

由於動態遊戲總是有某樣東西在螢光幕上到處移動，因此它必然是比較刺激的。若想讓圖形在螢光幕上移動，您只要一直不斷地在一連串稍為不同的位置上畫出相同的圖形即可辦到，其方法有二：第一種方法是在擦去舊位置之前先畫出新位置；在這種方法下，無論什麼時候一定是兩個物體邊

靠邊顯示在螢光幕上，這是古典的〝跳動的球〞（ Bouncing Ball ）副程式所使用的方法。在低解析度圖形中，這種方法有助於提高連續動作的幻覺。不過，在高解析度圖形中最好使用第二種方法：在畫出新位置之前先擦去舊位置，也就是說，在任何時刻只有一個物體在螢光幕上。

130 至 150 那幾列可畫出一輛坦克；我們是預先在方格紙上描繪出坦克然後再轉換爲X，Y座標的。

```
50 HGR
60 X=255
100 FOR HC=7 TO 0 STEP -7
120 HCOLOR= HC
125 Y=100
130 HPLOT X,Y TO X-4,Y-4 TO X-2,Y-6 TO
X+18,Y-6 TO X+22,Y-2 TO X+20,Y TO X,Y
140 HPLOT X+4,Y-6 TO X+6,Y-12 TO X+13,Y-12 TO
X+15,Y-6
150 HPLOT X+4,Y-9 TO X-12,Y-12
160 NEXT HC
170 X=X-PDL(0)/5
180 IF X<25 THEN GOTO 50
190 GOTO 100
200 END
```

該圖形是先以白色（ 100 那一列，HCOLOR＝7 ）畫出來，然後再以黑色（HCOLOR＝0 ）將它擦掉，而其位置係憑靠 160 那一列所構成的迴圈向前移動的，至於移動量之大小，卽移動之速度，則由 170 那一列中的 0 號控制器的位置所決定。

圖形本身是在130 ，140 和150 那幾列描繪出來的。它之所以能成爲一輛坦克是我們預先在方格紙上描繪出坦克的形狀，然後再轉換爲座標描繪出來的。您也可以隨意描繪自己的圖形，然後任意移動它。物體的大小和形狀是藉 130 ——150 那幾列設定的，而動作則由 170 那一列決定，因此，物體的大小和形狀與物體的移動可分開單獨改變，您可以隨意令任何物體按照您的企圖移動。

利用兩頁高解析度記憶頁的動態畫

要想獲得穩定不閃爍的動態畫，最好的方法就是同時利用那兩頁高解析度繪圖法，HGR 和HGR 2 。其秘訣就是在顯示一頁時偸偸地描繪另外一頁，使描繪與顯示交互進行。螢光幕上決不會有黑暗或閃爍的情況發生，因爲擦拭和重新描繪的過程觀察者是看不到的。

您必須在開始時利用HGR 和HGR 2 指令起用那兩頁螢光幕記憶體，在您利用這兩個繪圖 "軟體開關" (soft switch) 指令顯示它們之後， POKE−16300, 0 可顯示高解析度圖形的第 1 頁，而POKE − 16299, 0 可顯示第 2 頁。這些指令與HGR 和 HGR2 的作用一樣，只是它們不會把螢光幕清除爲黑色。繪圖軟體開關指令列於Applesoft 參考手册的第132 頁，您可以參閱。另外有兩個指令您需要用到，但Apple 忘了把它們包括在手册內：一個是POKE 230, 32 ，它係用來描繪第1 頁，但並不顯示它；另一個是 POKE 230, 64 ，它係用來描繪第 2 頁，但並不顯示它。描繪動態畫的程序大致如下：先顯示第 1 頁的圖形，然後跳到第 2 頁（但並不顯示它）清除舊圖形，再根據新位置描繪新圖形，然後顯示它，再回頭按同樣程序描繪第1 頁（先不顯示它），如此繼續下去。您必須很小心地盯住圖形的位置X；減 1 是要清除舊圖形，加 2 是要描繪新圖形。

您所需的程式步驟大體上如下所示：

```
POKE -16300, 0:REM DISPLAY PAGE 1
POKE230, 64:REM PLOT TO PAGE 2
(Erase & replot)
POKE-16299,0:REM DISPLAY PAGE 2
POKE230, 32:REM PLOT TO PAGE 1
(Erase & replot)
(Go back to the beginning & do it again)
```

下面是一段程式例，它可將一個小方塊從螢光幕的左邊移向右邊，當小方塊抵達螢光幕的右邊線時它會折返至左邊線。爲了使程式精簡，我們利用一個迴圈使變數PAGE在1與2之間交互變化。

```
90 REM ANIMATION-----
100 HGR:HGR2
110 X=10:Y=96:DX=2:REM DX = SPEED
120 FOR PAGE = 1 TO 2
130 POKE 230, 32 * PAGE: REM PLOT WITHOUT
DISPLAYING
140 X = X - DX: HCOLOR = 0: GOSUB 400: REM
ERASE OLD POSITION
150 IF X > 265 THEN 300: REM WRAPAROUND
160 X = X + 2 * DX: HCOLOR = 3: GOSUB 400:
REM PLOT NEW POSITION
170 POKE - 16301 + PAGE, 0: REM DISPLAY NEWLY
PLOTTED PAGE
180 NEXT PAGE
190 GOTO 120
300 REM WRAPAROUND-----
310 POKE 230, 32 * (3-PAGE): REM ERASE FROM
OTHER PAGE
320 X = X + DX: HCOLOR = 0: GOSUB 400
330 X = 2: REM PUT X ON LEFT OF SCREEN
340 GOTO 120
400 REM PLOTTING SUBROUTINE-----
410 HPLOT X, Y TO X + 10, Y TO X + 10, Y + 10
TO X, Y + 10 TO X, Y
420 RETURN
```

高解析度圖形，形狀表，與商用繪圖軟體

BASIC 雖然是一種非常方便使用的語言，但它天生地慢，因爲它是用傳譯，而不是用編譯的。〔傳譯器（ interpreter ）是在程式執行時才一列一列地轉換爲機器語言，而編譯器（ Compiler ）是在執行前把程式整個先翻譯成機器語言。〕因此，用BASIC 來做高速繪圖是很困難的。有一個解決辦法是利用形狀表（ shape table ）；這種形狀表實質上就是可描繪高解析度圖形，並在螢光幕上到處移動的高速機器語言副程式。Applesoft 參考手册雖然說明了如何去組成一個形狀表，但很不幸地，這是一件很沈悶的工作（至少我們可以這麼說）。然而又很幸運地，市面上有一些程式已經發表，可讓Apple 去做大多數較困難的工

作。John Figueres 曾經在Micro／Apple 一書中發表過一篇標題爲〝如何很容易，又正確地製作形狀表？〞（How to Do a Shape Table Easily and Correctly），而且在購買該書時附贈一張包含製作形狀表的程式和許多別的程式，在此鄭重推薦給您。

另外一個解決辦法是購買像Bill Budge 所寫的3－D Graphics （3－D繪圖程式）或Sirius Software公司的E－Z Draw商用套裝程式，這種套裝程式可讓您藉打入圖形的座標而定義二維或三維空間的形狀。一旦圖形建好之後您就可以利用BASIC 程式呼叫在該套裝軟體內編成的機器語言副程式以高速到處移動它們。還有兩個很優秀的套裝繪圖程式是美國加州San Rafael 城Stoneware 公司發展的Graphics Processing System （繪圖處理系統）和伊利諾州Geneva 城Penguin Software公司發展的The Graphics Magician （繪圖魔術師），Bring Murphy 先生曾在1983 年元月份的Creative Computing 雜誌上撰文討論過這兩個程式。

第三個解決辦法是自己學習Apple 機器語言。這當然不是初學者所能辦到的，您應該在考慮學習機器語言之前徹底學好BASIC ，況且有許多程式高手甚至從來沒有碰過機器語言。不過，如果您執意要學，坊間有許多書可協助您，您可以從Don 和Kurt Inman 合著的Apple Machine Language 開始。

圓，螺旋線與鐘面

要想了解如何以高解析度圖形繪製一個圓，有一些三角幾何觀念是有幫助的。圓可用一條旋轉的半徑尖端描繪出來，如圖1 所示。在某一瞬時，水平線與半徑之間的夾角爲A，半徑R 爲直角三角形OPQ 的斜邊，另外兩邊爲X和Y，標

著O的一個角位於該圓的圓心，從O點到P點的水平距離爲X（三角形的底邊），而從P點到Q點的垂直距離爲Y（三角形的垂直邊）。

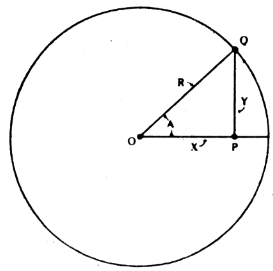
這也就是三角幾何學的緣起，記得：

Cos A＝底邊／斜邊

＝X／R

Sin A＝垂直邊／斜邊

＝Y／R



因此

X＝R・Cos (A)

Y＝R・Sin (A)

這些就是P點的X，Y座標。因爲圓係由所有旋轉半徑所掃過的點所組成，因此，當半徑掃一圈時（亦即夾角A由0度穩定增加到360度）只要描下所有P點的X，Y座標就可在高解析度螢光幕上畫出一個圓。（實際上，Applesoft不是用度來代表角度，而是用徑（radian）。一圈360度等於2π徑。）

下面這個副程式可以高解析度圖形畫一個圓：

```
10 HGR: HCOLOR = 3
20 PI = 3.14159
25 XC = 140: YC = 80: REM CENTER OF CIRCLE
30 R = 50: REM RADIUS OF CIRCLE
40 AMAX = 2 * PI
50 FOR A=0 TO AMAX STEP 0.2
70    X = XC + R * SIN (A)
80    Y = YC + R * COS (A)
90    IF A = 0 THEN HPLOT X, Y
100   HPLOT TO X, Y
110 NEXT A
```

這個圓的圓心位於點（140，80），其半徑爲50。您可以注意到這個圓的底部有一個很微小的間隙，這是因爲2π（50那一列內的AMAX）並不剛好是每

階間隔 0.2 的整倍數。

若想畫一組同心圓，只要加入下列幾列，然後重複執行這個副程式，並在每執行一圈之後令 R 遞增即可。

```
120 R = R + 10: IF R > 80 THEN END
130 GOTO 50
```

若想畫一條螺旋線，您在增加 R 時不能每執行一圈之後一階就增加很多，而是在 FOR A … NEXT A 迴圈裡持續小量增加。螺旋線其實就是在畫圓的時候持續穩定地增加其半徑。其主要形式有兩種：一種是阿基米得螺旋線（Archimedes spiral），其形狀像一只螺旋麪包，或捲起來之地毯。它是在畫螺旋線時持續加一個小常數至半徑而畫出來的。另一種是對數螺旋線（logarithmic spiral），其形狀像螺旋星雲，它是在畫螺旋線時持續乘（不是加）一個小常數至半徑而畫出來的。

若想描繪阿基米得螺旋線，可做下列修改：

```
DEL 120, 130

30 R = 3: REM INITIAL RADIUS OF SPIRAL
40 AMAX = 10 * PI
60 R = R + 0.2
```

而若想描繪對數螺旋線，可保留上面 30 和 40 那兩列，但是要將 60 那一列改為：

```
60 R = R * 1.02
```

對數螺旋線有很多有趣的特性，隨便畫一條半徑，您就會發現該半徑與螺旋線的每一圈所交的角度完全相同。對數螺旋線在大小上是無限制的；如果您取其中央的一小部份，比方說直徑 1 mm 將它隨意放大，它仍然會與該螺旋線的更大部份完全密合，即使直徑大於百萬公里。

讓我們再回到圓上：下面這個程式可以畫出帶有 12 個輻射狀小標記來代表時或方位的鐘面，並可畫出一支旋轉的指針來代表秒針或雷達螢光幕上的掃描線。

```
10 REM STOPWATCH OR RADAR SCAN-----
20 HGR: HCOLOR = 3
30 PI = 3.14159
40 R = 80: REM RADIUS
50 REM DRAW A CIRCLE-----
60 FOR A = 0 TO 2 * PI STEP 0.2
70   X = 140 + R * SIN (A)
80   Y =  80 - R * COS (A)
90   IF A = 0 THEN HPLOT X, Y
100  HPLOT TO X, Y
110 NEXT A
120 R = R - 2
130 REM DRAW 12 CLOCK MARKERS-----
140 FOR C = 0 TO 2 * PI STEP PI / 6
150 HPLOT 140 + R * SIN (C), R - R * COS (C)
TO 140 + (R - 10) * SIN (C), R - (R - 10) *
COS (C)
160 NEXT C
170 REM START SCAN-----
180 FOR B = 0 TO 2 * PI STEP 0.05
190   FOR I = 3 TO 0 STEP  -3
200     HCOLOR = I
210     HPLOT 140, R TO 140 + R * SIN (B), R
- R * COS (B)
220 NEXT I,B
230 GOTO 180
```

10 至 16 那幾列中描繪鐘面的原理很簡單，不必贅述。180 至 230 那幾列描繪旋轉的指針；首先以白色（HCOLOR = I = 3）畫出來，再以黑色（HCOLOR = I = 0）在描繪新指針位置之前擦去舊位置。該指針是順時鐘方向旋轉的，若要反時鐘方向旋轉，只要將 210 那一列中的第一個正號改為負號，亦即將 210 那一列改為：

```
210 HPLOT 140, R TO 140 - R * SIN (B), R - R
* COS (B)
```

旋轉速度是在 180 那一列中以每一階變化量 STEP 設定的（ = 0.05 ），若要使旋轉速度加倍，可將每一階變化量加倍為 0.1 。另外，您也可以用 0 號遊戲控制器來控制旋轉速度，只要做以下修改：

```
175 B = 0              (new line)
180 S = PDL (0) / 255 (was FOR B = 0 TO 2 *
PI STEP 0.05)
185 B = B + S          (new line)
220 NEXT I             (was NEXT I, B)
```

如果您願意圍著鐘面放入1至12的數字，您可以在TEXT （文字）模式中完成這件事，不必用高解析度圖形。其程式如下：

```
80 TEXT: HOME
90 R = 10: REM RADIUS
100 F = 1
110 GOSUB 200: REM PLOT CLOCK FACE
120 R = 8
130 F = 2
140 GOSUB 200: REM PLOT ROTATING ASTERISK
150 GOTO 140
190 REM SPECIFY 12 CLOCK FACE POSITIONS-----
200 FOR A = 5.2 TO 11 STEP 0.52
210   X = 20 + R * COS (A) * 1.25
220   Y = 20 + R * SIN (A)
230   HTAB X: VTAB Y
240   ON F GOSUB 300, 350
260 NEXT A
270 RETURN
300 REM PRINT DIGITS-----
310 PRINT INT(2 * A - 9.3): REM 1 THROUGH 12
320 RETURN
350 REM PRINT MOVING ASTERISK-----
360 PRINT "*";
370 K = PEEK (-16336) + PEEK (-16336);REM
CLICK
380 FOR T = 1 TO 255 - PDL (0): NEXT T
410 HTAB X: VTAB Y
420 PRINT "   ";REM ERASE OLD ASTERISK
430 RETURN
```

注意看看這程式與前一個程式有多像。210和220那兩列中的R＊SIN(A)和R＊COS(A)完全一樣，只是不再指定到那裡HPLOT一點，而是利用230和410兩列中的HTAB X：VTAB Y指令去PRINT一個字元。

旋轉速度是藉0號控制器的讀數（ 380 那一列）控制的。您也可以將這種原理應用在賭博遊戲中，令一個星號先快速旋轉，然後漸漸慢下來，最後停在一個1至12之間的亂數（ random number ）上，就好像您在遊樂場所看到的輪盤遊戲（ wheel of fortune ），只是數目由36個減爲12個而已。若想製作輪盤遊

戲可做下列修改：

```
250 IF TMAX > 100 THEN GOTO 450

380 FOR T = 1 TO TMAX: NEXT T
390 TMAX = TMAX + RND(6)
400 IF TMAX > 100 THEN RETURN

450 TMAX = 0
460 VTAB 23: HTAB 1: PRINT "HIT ANY KEY FOR
ANOTHER RUN"
470 GET H$
480 GOTO 80
```

讓這個副程式多執行幾次，它就會像模擬一只 12 面的骰子在產生亂數一樣。事實上，您會發現它並不如您預期的一樣"亂"，它好像會比較偏好一些數目。試試看將 250 和 400 那兩列改為

```
250 IF TMAX > 200 THEN GOTO 450
400 IF TMAX > 200 THEN RETURN
```

看看會不會好一些。如果不行，您可以自己試試其他的解法。

多邊形，星形與旋形圖(Squiral)

電腦實際上是絕對畫不出正圓的：它只是藉描繪一個極多邊的多邊形來近似一個圓。若想描繪一個正規的多邊形，我們可以在一個圓上選取一些等距的點，然後用直線把它們連接起來。

```
100 REM POLYGON-----
110 PI = 3.14159
120 XC = 140: YC = 96: REM CENTER OF POLYGON
130 R = YC - 3: REM RADIUS
140 TEXT: HOME
150 INPUT "POLYGON: HOW MANY SIDES ";N
160 S = 2 * PI/N: REM STEP SIZE
200 HGR2: HCOLOR = 3
210 FOR A = 0 TO 2 * PI + S STEP S
220 X = XC + R * SIN (A)
230 Y = YC + R * COS (A)
240 IF A = 0 THEN HPLOT X, Y
250 NEXT A
```

除了使用者必須決定多邊形的邊數外，本程式與畫圓的程式完全相同。如果

您 INPUT 一個很大的數目給N，譬如說 100 ，這程式就會畫出一個接近完美的圓。您可以當作一個練習，將這程式加以修改使它能夠描繪多邊螺旋線（ polygonal spiral ）。

如果您在連接多邊形的邊線時每隔一點連接一線，您將會得到一個很美麗的圖案，這種圖案可作爲測試電腦繪圖品質的標準方法。其程式如下：

```
80 REM STAR POLYGON-----
90 REM SET THE VARIABLES-----
100 PI = 3.14159
110 XC = 140: YC = 96: REM CENTER
120 R = YC - 3: REM RADIUS
130 TEXT: HOME
140 DIM X(50), Y(50)
150 FOR N = 4 TO 20: REM N = NO. OF SIDES
160 J = 0
170 S = 2 * PI/N: REM STEP SIZE
180 HGR2: HCOLOR = 3
190 REM CALCULATE CO-ORDINATES-----
200 FOR A = 0 TO 2 * PI + S STEP S
210 J = J + 1
220 X(J) = XC + R * SIN (A)
230 Y(J) = YC + R * COS (A)
240 NEXT A
250 JMAX = J
260 REM PLOT THE FIGURE-----
270 FOR J = 1 TO JMAX
320 FOR K = 1 TO J
330 HPLOT X(J), Y(J) TO X(K), Y(K)
340 NEXT K, J
350 NEXT N
```

本程式所繪製的是整整一系列的星狀多邊形，從正方形（ N＝4 ）一直到20邊形。您若想在表演的時候讓程式不斷地循環，可加入

```
360 GOTO 150
```

另一方面，如果您只想畫一個星狀多邊形以便有空時才檢查它，您可以做以下修改：

```
DEL 350, 360
150 INPUT "STAR POLYGON: HOW MANY SIDES ";N
```

讓我們假設您已經修改好上面的Polygon（多邊形）程式〔不是 Star Polygon（星狀多邊形）〕，可以畫出多邊形螺旋線了。您可以進一步看看1982年

12 月份的Creative Computing 雜誌中第274 頁Fred Gunther 先生發表的一篇既好又短的程式，它可繪製“旋形圖”（ squiral ）。

```
5 ONERR GOTO 30
10 TEXT: HOME
20 PI = 3.14159: W = PI/180
30 INPUT "SQUIRAL: ANGLE, RADIUS, INCREMENT
";ANGLE, R, INCR
40 HGR2: X= 140: Y = 96: HCOLOR = 3: A =
ANGLE
50 XN = X + R * SIN (W * A): YN = Y + R * COS
(W * A)
60 HPLOT X, Y TO XN, YN
70 R = R + INCR
80 A = A + ANGLE
90 X = XN: Y = YN
100 GOTO 50
```

什麼是“旋形圖”呢？它是一個一邊擴大一邊微微旋轉的多邊螺旋線。如果該螺旋線的角度正好等於120 ， 90 或 60 度，則您會分別得到一個三角形，正方形或六邊形的螺旋線；但是，如果角度爲118 ，120 ， 123 或 89 ， 90 ， 91 或 59 ， 60 ， 61 ，… 等數值，則您將會得到很有趣的結果。如果您願意，您還可以加一些指令爲旋形圖塡上彩色背景（試試HCOLOR＝2：HPLOT 0，0：CAIL 62454）。

若想讓旋形圖以隨機選擇的數值連續顯示，可將 30 那一列改爲：

```
30 ANGLE = 360 * RND (1): R = 5 * RND(2):
INCR = 5 * RND (3)
```

注意，5 那一列是懶人的錯誤捕捉器（ error trap ）， 當旋形圖畫到螢光幕邊緣時程式便會注意到錯誤而跳回 30 那一列重頭再畫旋形圖，不是光印出 ILLEGAL QUANTITY 訊息就停止了。

旋形圖如果以一種稱爲 Logo 的語言來畫是非常容易的，它可讓使用者控制一個可在螢光幕上到處移動並留下軌跡的電子小海龜（參閱相關書籍）。Logo 的作者 Seymour Papert 先生寫過一本很吸引人的書，書名爲 Mindstorms：

Children, Computers and Powerful Ideas: All About Logo — How It Was Invented and How It Works (Basic Books, 1980)， 很值得一讀。

平均值與插值法

要求出兩個數目，Y1 與 Y2 ，的平均值是很容易的：

```
10 INPUT "Y1=";Y1
20 INPUT "Y2=";Y2
30 MEAN = (Y1 + Y2) / 2
40 PRINT "AVERAGE OF Y1 AND Y2 = ";MEAN
```

平均值其實就是 Y1 和 Y2 之間的中點。但是，有時候，人們並不想找出 Y1 和 Y2 的中間數目，而是想在 Y1 和 Y2 之間插入 10 個間隔相等的數目，這就是插值法（ interpolation ）。

下面是做法之一：

```
10 Y1 = 5
20 Y2 = 35
30 FOR J = 0 TO 10
40   Y(J) = J/10 * Y2 + (10 - J)/10 * Y1
50   PRINT J, Y(J)
60 NEXT J

RUN

0        5
1        8
2       11
3       14
4       17
5       20
6       23
7       26
8       29
9       32
10      35
```

請看 50 那一列，Y(J)等於一份 Y2 加一份 Y1 ： 亦即，Y2 的$\frac{1}{10}$加 Y1 的 $\frac{9}{10}$，然後 Y2 的$\frac{2}{10}$加 Y1 的$\frac{8}{10}$，…，以此類推。當 J 由 0 至 10 愈來愈大時 $\frac{J}{10}$也愈來愈大，使得 Y2 漸漸增加，而（ 10－J ）／10愈來愈小，使得 Y1 漸漸變小。因此，

Y(J)的數值將從 Y1 開始以相等間隔趨近 Y2 。

下面是一個雙重插值程式，可同時在兩對數目，X1 與 X2 和 Y1 與 Y2 ，之間插值。

```
20 TEXT:HOME:VTAB 10
30 INPUT "X1="; X1
40 INPUT "X2="; X2
50 INPUT "Y1="; Y1
60 INPUT "Y2="; Y2
70 KX = ABS (X1 - X2)
80 KY = ABS (Y1 - Y2)
90 K= KX
100 IF KY > KX THEN K = KY
110 PRINT "#", "X", "Y"
120 FOR J = 0 TO K
130   X(J) = J/K * X2 + (K - J)/K * X1
140   Y(J) = J/K * Y2 + (K - J)/K * Y1
150   PRINT J, X(J), Y(J)
160 NEXT J

RUN

X1=10
X2=15
Y1=50
Y2=60

#            X            Y
0            10           50
1            10.5         51
2            11           52
3            11.5         53
4            12           54
5            12.5         55
6            13           56
7            13.5         57
8            14           58
9            14.5         59
10           15           60
```

20 至 60 那幾列接受 INPUT 數目 X1 ， X2 ， Y1 和 Y2 。注意， X2 只比 X1 多 5 ，而 Y2 比 Y1 多 10 。 70 至 100 那幾列偵查這一點，並決定在 X1 與 X2 和 Y1 與 Y2 之間插入 10 點（不是 5 點）。 130 和 140 那兩列進行計算，而 150 那一列將它們 PRINT 出來。

這樣做有什麼用處呢？請先做下列修改：

```
10 DIM X(300), Y(300)
110 HGR: HCOLOR = 3
150 HPLOT X(J), Y(J)
```

您就會發現當您 INPUT X和Y兩對點時這副程式便會在它們之間HPLOT 一條高解析度線，這等於您又發明了一個 Apple 原已用機器語言建在內部的 HPLOT To X，Y 指令的做法。

您可能已注意到Apple 中沒有對應於HPLOT 指令而可在任何兩個點X，Y間畫一條直線的低解析度指令——但是，現在您有了！您可以利用這個副程式在低解析度圖形中畫直線，只要做下列修改：

```
110 GR: COLOR = 15
150 PLOT X(J), Y(J)
```

爬行的字幕

若想PRINT 出一段可橫向“爬”過螢光幕而眼睛又跟得上的訊息，您可以利用下面的程式：

```
100 TEXT:HOME
110 A$ = "WATCH THIS MESSAGE...."
120 L = LEN(A$)
130 A$ = MID$(A$, 2) +LEFT$(A$, 1)
140 VTAB 21
150 HTAB 20 - L / 2
160 PRINT A$;
170 FOR T = 1 TO PDL (0): NEXT T
180 GOTO 130
```

130 那一列靠下面方法令訊息“爬行”：輪流將訊息的第一個字取出，然後移到該訊息的最後面，就好像洗牌時將第一張牌放到該付牌的底部一樣。(MID$(A$,N)得到的是由 A$ 字串中第N個位置開始，一直到 A$最後一個字元的分字串(substring)，而 LEFT$(A$, N) 得到的是 A$最左邊的N個字元。) 150 那一列自動爲HTAB選一個數值將該訊息置於水平方向的中央。170 那一列在0號控制器的控制之下決定訊息“爬行”的速度(從快到慢)。

令同一段訊息（或兩段不同訊息）反方向爬行，一段向右而另一段向左，並不會太困難：

```
50 A$ = "HELLO, YOU GOOD-LOOKING GENIUS...."
60 L = LEN (A$)
70 B$ = A$
100 TEXT: HOME
110 A$ = RIGHT$ (A$, L-1) + LEFT$ (A$, 1)
120 B$ = RIGHT$ (B$, 1) + LEFT$ (B$, L-1)
130 VTAB 6: HTAB 20 - L/2
140 PRINT A$
150 VTAB 12: HTAB 20 - L/2
160 PRINT B$
170 GOTO 110
```

亂數

亂數產生器（Random Number Generator）是BASIC中最有用的特點之一，因爲它可讓您模擬隨機發生的事件，如投擲輔幣或骰子，或複雜如放射性分子隨機分裂，以及醉漢隨機蹣跚等之類的事件。

Applesoft與整數BASIC的亂數函數稍爲有點不同。讓我們先看看整數BASIC（如果您從來不在Apple上使用整數BASIC，您可以跳過本節。）

整數BASIC中的亂數函數

若要確定我們是否處於整數BASIC中，可以看每一列最前面的提示信號（prompt）是否爲“＞”。

下面函數

```
>10 X = RND (10)
```

將X隨機設定爲一個介於 0 與 9 之間的整數。若想得到 0 與 10 之間的整數，可利用

```
>10 X = RND (11)
```

因此

```
>10 X - RND (10) + 1
```

可得到 1 與 10 之間的整數。擲一粒骰子可獲得的數目介於 1 與 6 之間，因此您可以利用

```
>10 D = RND (6) + 1
```

擲一對骰子可獲得的數目介於 2 與 12 之間，因此您可以試試

```
>10 D1 = RND (6) + 1
>20 D2 = RND (6) + 1
>30 D = D1 + D2
>40 PRINT D
```

注意，您雖然也可以利用下式來得到 2 與 12 之間的亂數

```
>10 D = RND (11) + 2
```

但是，這是兩回事，因爲此刻介於 2 與 12 之間的任何數目好像面朝上的機會都是一樣多，對一對眞正的骰子，或上面那四列程式，7 是最容易出現的，其次是 6 或 8 ，再次爲 5 或 9 ，…以此類推；其原因將在“頻率分佈”中加以說明。

亂數對繪圖是很有用的。在低解析度圖形中，那十六種顏色每一種顏色分別被指定 1 個 0（黑色）至 15（白色）的數目。若想隨機選擇顏色（包括黑色），您需要獲得 0 與 15 之間的亂數。

```
>10 C = RND (16)
>20 COLOR = C
```

若想排除黑色，可選擇 1 至 15 之間的亂數：

```
>10 C = RND (15) + 1
>20 COLOR = C
```

您也可以隨機選擇X和Y的座標。在低解析度圖形中，水平方向和垂直方向的座標都介於 0與 39 之間。試著把下面程式打進去，再RUN看看：

```
>10 GR
>20 C = RND (15) + 1
>30 COLOR = C
>40 X = RND (40)
>50 Y = RND (40)
>60 PLOT X, Y
>70 GOTO 20
```

這個小程式會把五彩碎紙隨機散佈在整個螢光幕上。若想脫出程式，可按 Ctrl — C 或 RESET 。

若想在隨機位置上描繪直線，您可以利用下面指令形式：

```
>10 HLIN A, B AT Y
```

但是要記得B必須大於A。因此，若僅僅把A設定爲RND(40)，並且把B設定爲RND(40)是不夠的，因爲大約有一半時間B會比A小；那時，您就會看到SYNTAX ERROR（語法錯誤）的訊息。解決的辦法是利用

```
>10 GR
>20 COLOR = RND (15) + 1
>30 A = RND(40)
>40 B = A + RND(40 - A)
>50 Y = RND (40)
>60 HLIN A, B AT Y
>70 GOTO 20
```

40 那一列令 B 大於 A，而 RND(40 — A) 則確保 B 不會超過 39 。

Applesoft的亂數函數

在 Applesoft 中，X = RND(3) 會產生一個介於 0 與 1 之間亂數，小括弧中的引數（ argument ）可為任意正數，因此使用 RND(1)， RND(7.45) 或 RND (359) 一樣好。如果您在程式中已經先獲得變數 Q 的數值，則引數也可以是變數，例如 RND(Q)。（如果沒有預先得到 Q，則 Q 將仍為系統設定之零，因此 RND (Q) 相當於 RND(0)，那將會產生前面用過的最後一個亂數（下面會說明）。

若想獲得大一點的亂數，您可以乘上一個數目，像 X = 10 * RND(3) 可提供 0 與 9.9999 …之間的數目（繼續研讀您就會明白。）

修截亂數

Applesoft 產生的亂數有一長串小數位數，您只要利用 INT 函數就可將它們截為次小的整數：

```
10 R = 10 * RND (1)
20 R = INT (R)
```

注意，10 那一列會產生一個 0 與9.99999之間的數目，經過 20 那一列截取後可得到 0 與 9 之間的整數一不是 0 與 10 之間。若想獲得一個 0 與 10 之間的數目可利用：

```
100 R = INT (10 * RND (1) + 0.5)
```

加上+ 0.5 可將 R 修截為最接近它的整數，譬如 5.49 會截為 5 ，而 5.51將

會變爲6，因此任何介於9.51與9.999之間的亂數都會被修整爲10，您也就可以獲得相等出現次數的0，1，2，… 10。如果您想要的是1與10（不是0與10）之間的亂數，可利用：

```
100 R = INT (10 * RND (1)) + 1
```

若想將一個帶有小數的亂數修截至小數點後兩位，可用：

```
10 R1 = 10 * RND (1)
20 R2 = INT (R1 * 100)/100
30 PRINT R1,R2

RUN

4.966275083       4.96
```

您可以將這個程式加以修改，使它能夠將一個亂數修截至所需的小數位數：

```
10 INPUT "HOW MANY DECIMAL PLACES ?  ";DP
20 E = 10 ^ DP
30 R = RND (1)
40 R = INT (R * E)/E
50 PRINT R
60 GOTO 10

RUN

HOW MANY DECIMAL PLACES ?  1
3.7

HOW MANY DECIMAL PLACES ?  2
8.51
HOW MANY DECIMAL PLACES ?  6
2.864579
HOW MANY DECIMAL PLACES ?  0
0
....
```

亂數可重複嗎？

RND函數的用法包含在Applesoft使用手册的第102頁和159頁。RND(5)

或RND(8)雖然都會產生一個介於0與1之間的隨機實數，不過只要引數為正，RND每次所得到的亂數都不會相同（引數就是RND後面小括號中的數目）。請看下面例子：

```
20 FOR J = 1 TO 6
30 R = RND (3)
40 R = INT (R * 100)/100:REM TRUNCATE TO 2
DECIMAL PLACES
50 PRINT R,
60 NEXT J

RUN

.78             .24             .13
.62             .53             .09

RUN

.36             .43             .92
.57             .48             .61
```

每次執行所產生的亂數數列都不一樣。

然而，如果利用某個負數作為引數，如RND(−3)或RND(−5)，則隨後由引數（RND(9)或RND(2)等）所產生的亂數數列將會每次都相同，就好像有一個固定的亂數數列建在Apple內一樣。負引數如果不同，所產生的亂數數列也不會相同。在RND中使用負引數的主要理由是作為產生一組重複出現的亂數數列的種子；這種數列在為使用RND的程式除錯時特別有用。此外，這種數列在製作某種可見形式的配對亂數組合時也很有用的。

讓我們將10那一列加入上面的程式，並重新RUN它：

```
10 S = RND (-2): REM ENSURE FIXED SEQUENCE OF
RANDOM NUMBERS
20 FOR J = 1 TO 6
30 R = RND (3)
40 R = INT (R * 100)/100
50 PRINT R,
60 NEXT J

RUN

.27              .62              .36
.3               .57              .04

RUN

.27              .62              .36
.3               .57              .04
```

每次 RUN 這個程式都會產生完全相同的亂數數列。

現在，將 10 那一列改為：

```
10 S = RND (-5)
```

然後再 RUN 個兩三次，您將會得到一組不同於上次的數列（因為我們已將－2換成－5），但是如果您再 RUN 這個程式，它所產生的數列仍將會一樣：

```
RUN

.4               .46              .38
.23              .14              .41

RUN

.4               .46              .38
.23              .14              .41
```

RND(0) 可以獲得前面所產生過的最後一個亂數（CLEAR 和 NEW 對它沒有影響）。有時候，利用 RND(0) 比將最後一個亂數指定給某個變數再存起來還要容易。試試看將它加入上面的程式：

```
70 PRINT RND (0)

RUN

.4              .46             .38
.23             .14             .41
.416384865
```

同樣的數列又重新出現，但在最後面以RND(0)取得前面產生過的最後一個亂數，這亂數保有未經修截的數值.416384865。

零與壹

若想模擬諸如投擲輔幣之類只有兩種可能結果的機率事件，您可以利用下法隨機產生0與1：

```
10 X = 0
20 IF RND (1) > 0.5 THEN X = 1
```

比較漂亮的方法是利用可判斷某個陳述式(statement)眞僞的邏輯函數。例如，在下式中：

```
10 X = (A > B)
```

變數X稱爲"眞值函數"(truth function)。如果A大於B，則其值爲1(="眞")；如果A等於或小於B，則其值爲0(="僞")。眞值函數又稱爲"布林運算式"(Boolean expression)，用以紀念邏輯學家布林(Boole)。假設我們用亂數RND(1)來取代A，則：

```
10 A = RND (1)
20 X = (A > 0.5)
30 PRINT X;
40 GOTO 10

RUN

01110110100100.....
```

長時間來看，這個程式會產生同等數目的0與1（對輔幣而言，相當於“人頭”與“梅花”）。若想使0的數目為1的兩倍，可將20那一列改為：

```
20 X = (A > 0.66)
```

重新執行程式，並注意其結果。

```
RUN
10001011000100010000.....
```

設定上下限

下面這個小程式可產生介於隨意設定之最大和最小數目之間的亂數：

```
10 INPUT "MIN ";MIN
20 INPUT "MAX ";MAX
30 R = MIN + (MAX - MIN) * RND (7)
40 PRINT INT (R + 0.5)
50 GOTO 10

RUN

MIN ?100
MAX ?200
138
MIN ?100
MAX ?200
104
MIN ?100
MAX ?200
169
MIN ?10
MAX ?20
11
MIN ?10
MAX ?20
19
....
```

隨機時間延遲

上述方法同樣可用來產生帶有預置上下限的任意隨機時段（ time interva

其程式如下：

```
10 INPUT "MIN ";MIN
20 INPUT "MAX ";MAX
30 TC = 700: REM CONVERTS TIME INTERVAL INTO
SECONDS
40 PRINT "HIT ANY KEY TO START TIME INTERVAL"
50 GET H$: HOME
60 TMAX = TC * (MIN + (MAX - MIN) * RND (1))
70 FOR T = 1 TO TMAX: NEXT T: REM TIMING LOOP
80 CALL - 1059: REM BEEP
90 GOTO 50: REM DO IT AGAIN
```

本程式之作用好像一個橡膠球在一個假想的空間內亂跳，直到跳進一扇由 10 和 20 那兩列中輸入的 MIN 和 MAX（以時間來表示）所定義的假想窗框內。窗戶打開的大小決定平均起來球從跳動至終止的時間長短。70 那一列是一個計時迴圈（timing loop），利用電腦在內部自行計數至一個數目（TMAX）而達到延時的目的，進而決定時段的長短。TMAX 是在 60 那一列內被設定為一個介於 MAX 上限和 MIN 下限之間的隨機秒數。筆者曾利用一只碼錶來做實驗，發覺筆者的 Apple 一秒鐘約可執行 700 次計時迴圈，因此在 60 那一列的運算式內乘上一個 700 的時間常數（TC）。如果您的機器有些許差異，您可以利用碼錶來校正程式。

東倒西歪(Random Walk)

隨意投擲的一枚輔幣 1000 次大約可以各得一半次數的人頭和梅花，每一次投擲完全是一件獨立的事件，絲毫不會受前一次投擲的影響。不過，賭徒中流傳一個古老的錯誤說法：一枚隨意投擲的輔幣如果已經出現出 20 次人頭，則下一次必定出現梅花。事實上，輔幣是沒有記憶力的，不管出現幾次人頭，下一次出現梅花的概率仍然等於 0.5 。（我們姑且不談有這種迷信想法而在投擲輔幣時做出某

種刻意行爲的人。我們假定的是一枚完全不是刻意投擲的輔幣。）因此，投擲輔幣或旋轉輪盤完全是不相干的獨立隨機事件。

不過在另一方面，有一類隨機事件的結果確實是與前面發生的事有關（數學家稱這種爲“ Markov chain ”），譬如說醉漢的步伐雖然完全隨機，其方向也不可預料，但我們仍然可以知道他的下一步一定只離開他現在站立的位置一步遠。氣體的分子正像一個很小的醉漢一樣，受其他分子亂衝亂撞，歪歪扭扭的跳著布朗捷克舞（ Brownian jig ），但它的下一步大大地與它現在的位置有關。如果您從 100 元中拿出 1 元來賭一枚輔幣出現人頭，我們雖然根本無法預料您會贏或輸，但是我們仍然可以預測您的財務狀況：我們知道不管您輸贏，您最後將會有 $ 101 元或 $ 99 元，亦即 $ 100 加或減 1 元。

由上面例子顯示出一個變動量一醉漢或氣體分子的位置，或者賭徒的錢財一雖然隨機變動，但它的值與兩件事有關：(1)它的現值（已知），(2)下一步會怎樣（未知）；換句話說，該變數是在“東倒西歪”。

東倒西歪—加一或減一

我們由上面知道眞値函數（ RND(1) > 0.5 ）的値是 1 或 0 完全根據隨機變數 RND(1) 大於 0.5 或小於 0.5 ，這種眞値函數可以控制變數 X 的數値，令它隨機選擇加一或減一。

```
10 X = 0
20 R = RND (5)
30 X = X + 2 * (R > 0.5) - 1
40 PRINT X,
50 GOTO 20

RUN

1              0              -1
1              2               3
4              2               1
2              1               0
-1             -2....
```

在 30 那一列中，如果括弧中運算式爲眞（大約有一半時間）則 X 的數值加 2，否則加 0，亦卽其餘時間維持不變；因此，X 不是增加 2 就是不變。30 那一列中最後面“－1”的目的是將 X 的變化量改爲＋1 或－1。

這種方法可用來令一個低解析度點在螢光幕上做布朗運動（Browian motion）：

```
10 GR
20 C =INT (16*RND(1))
30 FOR HUE = C TO 0 STEP 0
40 COLOR = HUE
50 X = X + 2 * (RND(1) > 0.5) -1
60 Y = Y + 2 * (RND(1) > 0.5) -1
70 IF X < 0 THEN X = 0
80 IF Y < 0 THEN Y = 0
90  IF X > 39 THEN X = 39
100 IF Y > 39 THEN Y = 39
110 PLOT X, Y
120 NEXT HUE
130 GOTO 20
```

本副程式在圖形顯示程式“蝴蝶”（Butterfly 1 和 Butterfly 2）中都曾用到。

若想讓 X 每一次經過迴圈後的增量爲－1，0 或＋1，可將 30 那一列改爲：

```
30 X = X + (R > 0.67) - (R < 0.33)
```

有三分之一機會 R＞0.67 將爲眞，因此 X 會增加 1，有三分之一機會 R＜0.33 將爲眞，因此 X 會減 1，而另外三分之一機會前兩項都不爲眞，因此 X 不會改變。我們可以利用這種方法修改“蝴蝶”程式：

```
45 R = RND(1): S = RND(3)
50 X = X + (R > 0.67) - (R < 0.33)
60 Y = Y + (S > 0.67) - (S < 0.33)
```

不等步伐大小的東倒西歪

若想使X以正或負與不等的間隔變動，可利用：

```
20 X = 10 * RND(1) - 10 * RND(1)
30 PRINT INT (X); " ",
40 GOTO 20
RUN

3 -10 -4 6 0 -4 -1 -4 -2 -8 1 -1 -5 -7 0 10
8.....
```

在20那一列中，前面的10＊RND(1)會產生一個介於0與10之間的亂數，後面的10＊RND(1)也會產生一個介於0與10之間的不同亂數，兩者相減之後會得到一個介於＋10與－10之間的亂數。

若想將這些一起納入〝東倒西歪〞程式令X的數值與它自己的前一個數值有關，可利用：

```
10 X = 100
20 X = X + 10 * RND (1) - 10 * RND (1)
30 PRINT INT (X); " ";
40 GOTO 20

RUN

103 98 94 95 92 100 100 101 102 99 100 101
107 108 100 99 94 95 92 90 87.....
```

下面是一個高解析度東倒西歪程式：

```
10 HGR: HCOLOR = 3
20 X = 0: Y = 80
30 HPLOT X, Y
40 X = X + 1
50 Y = Y + 10 * RND (1) - 10 * RND (1)
60 HPLOT TO X, Y
70 IF X = 279 THEN END
80 GOTO 40
```

這個副程式已被用於策略遊戲〝股票市場〞（Stock Market）中來描繪股票價格的不規則變化。有一些分析家相信市場趨勢是隨機的，但有一些人則

相信股票價格的變化是可預測的，亦即非隨機的，並據以賭他們的財富。

下面是前面程式的二維空間版本，在這種程式裡高解析度點可在X與Y方向隨機移動：

```
10 HGR: HCOLOR = 3
20 X = 140: Y = 80
30 HPLOT X, Y
40 X = X + 5 * RND(1) - 5 * RND(1)
50 Y = Y + 5 * RND(1) - 5 * RND(1)
60 HPLOT TO X, Y
70 GOTO 40
```

這種程式可以很清楚地描述氣體分子的布朗運動，已被用於教育／科學程式系列的〃東倒西歪〃（Random Walk）程式中。

濾平(Smoothing)

從上面的股票市場浮動副程式可以看出緩慢，長期的趨勢以及快速，每日的股價變動。航海家可能把它看成一股夾著漣波和碎浪的緩慢巨浪，而磁帶記錄器或錄音機技術員可能認爲它們是雜波。對電機工程師而言，它們很可能只是某種低頻和高頻的波形而已。然而，無論應用在那裡，觀念是完全相同的。

我們可以利用下面程式濾除高頻變化，僅保留低頻變化：

```
10 Y1 = 60: Y2 = 80
20 R = 5 * RND(1) - 5 * RND(1)
30 Y1 = Y1 + R
40 INCR = (INCR + R)/2
50 Y2 = Y2 + INCR
60 GOTO 20
```

30 那一列令Y1 隨機變化，40和50 那兩列令Y2 也變化，但中間變數I-NCR就好像慣性負載（inertial load）一樣減弱Y2 內的快速變化，但仍讓它保有像Y1 一樣長期的趨勢。注意 20 那一列，R在上下限＋10與－10之間到處亂跳，該列毫無記憶力，因此，每一次經過迴圈所獲得的R值與前一次經過迴圈所獲得的R值完全無關。不過，在 30 那一列中，Y1 的數值則視前面的數值而

定，因爲Y1（新值）等於Y1（舊值，上一次經過迴圈的記憶值）+R。在40和 50 那兩列中，Y2 的數值又多一層限制，因爲它的變化率不僅受 Y2 舊值的影響，還受 INCR舊值的影響，濾平 Y2 是藉平均 Y2 與其舊值而達到目的的（想想看！）。

下面是一個以更通用形式納入上面 40 和 50 那兩列的高解析度程式（請注意 70 和 80 那兩列）。螢光幕下面的波狀線（Y2）是把上面的波狀線（Y1）濾平而得的。

```
10 INPUT "HOW MUCH SMOOTHING (0--10)? ";K
20 HGR2: HCOLOR=3
30 Y1=60: Y2=80
40 FOR X = 0 TO 279
50   R = 5 * RND(1) - 5 * RND(1)
60   Y1 = Y1 + R
70   INCR = (K * INCR + R)/(K + 1)
80   Y2 = Y2 + INCR
90   HPLOT X, Y1
100  HPLOT X, Y2
110 NEXT X
120 HOME: VTAB 22
```

使用者在 10 那一列中 INPUT 的變數K係用來設定濾平量的阻尼常數（damping constant）。如果K設爲0，則 70 那一列變爲 INCR＝R，因此 Y2＝Y2＋R，Y2的變化量與 Y1 相同，因爲根本沒有濾平作用。如果K設爲1，則 INCR＝INCR＋R，與原來的濾平方法相同。如果K設爲其他值，譬如 9，則

```
70 INCR = (9 * INCR + R)/10
80 Y2 = Y2 + INCR
```

阻尼量就大大提高了，因爲 Y2 的新值是由 90%的 INCR 舊值所決定，只有10%由新的 R值決定。因此，INCR的作用就好像固定負載或阻尼器一樣減弱Y2數值中的隨機變化量，提供短期的穩定性。就長期而言，Y2 仍然跟得上 Y1 的變化趨勢，但短期而言，Y2 內的變化被〝熨平〞了。

未被濾平的波狀線 Y1 用於策略遊戲 Stock Market（股票市場）中來模擬

股價的變化，而經過濾平的波狀線 Y2 則用於策略遊戲 Running On Empty（爬坡）中來模擬崎嶇不平的路面。道路工程師通常都需要花費一筆龐大的經費於輾平可能會使車輛產生不舒服振動的起伏路面，但他們通常無法負担剷平可能影響行車舒適程度的大山丘，這種長距離或短距離的起伏路面都可用時間來模擬。磁帶記錄器或錄音機的捲帶速度有時也會有人們極不希望的隨機變動；相當於道路上某一點突然起伏的快速變化會產生高頻的尖刺聲音，而低速變化也會產生同樣不悅的聲音。

您也可以實驗一下負阻尼的情況。將 50 那一列改爲

```
50 R = 2 * RND (1) - 2 * RND (1)
```

然後執行它，並輸入一個大約－0.505 的阻尼係數。注意看螢光幕，您將會看到高頻變化不僅沒被濾平，反而漸漸增大產生振盪，甚至會超出螢光幕而迫使程式中斷。同樣地，這種現象在聲音方面也有類似情形。如果麥克風偶而太接近廣播系統的揚聲器，它會從揚聲器撿拾到極微弱的聲音，將它放大，……，您就會聽到所謂的吼聲（howling）或正回授（positive feedback）。這是一個由負阻尼產生振盪的例子，也就是人們常說的正回授。

自然界也常常可以看到濾平的例子，如果太陽所產生的熱量 R 由於雲層的變化而引起隨機性的變化，則空氣溫度（Y1）也就會急速變化。不過，湖水溫度（Y2）變化就緩慢得多了，完全看它當時的溫度與太陽光照射湖面（R）的〞熱慣性〞（INCR）而定，這也就是爲什麼沿海城市的氣候比內陸城市穩定的原因。同樣地，跑車駕駛員踩煞車或油門（R）的動作會快速反應在跑車的速度（Y1）上，而重卡車的速度（Y2）則多半決定於它當時的動量（INCR），所以卡車要花較長的時間加速；同樣地，也要花較長的時間將它煞住。

不等頻率分佈的亂數

下面函數

```
10 R = INT (10 * RND (1))
```

會產生 0 與 9 之間的整數，如果您 RUN 它 1000 次，每一個數目出現的機會大約都是 100 次。這種相等頻率的特性稱為〞矩形頻率分佈〞(rectangular frequency distribution)，因為如果您沿著圖形底部標下可能出現的數目（此處為 0 至 9 ），而在縱軸方向描繪各個數目出現的次數，您會得到一個長方形，其高度為 100 個單位。

這種分佈沒什麼不好，但您有時候可能也會想要別種頻率分佈。如果您描下您在街上遇到的頭 1000 人的身高，您就會發現身高的分佈並不是長方形，因為並非每個人都具有同樣的高度，反而是中等身材的人要比特別高或矮的人更普遍；巨人與侏儒非常少，大多數人的身高平均集中在 5 呎 8 吋而呈現鐘形曲線。這種鐘形曲線稱為“常態分佈”(normal distribution)；又稱為“高斯分佈”(Gaussian distribution)，以紀念最先研究它，人稱數學王子的德國人 Karl Friedrich Gauss 。

這種鐘形曲線在自然界是很普遍的－遠比矩形分佈來得普遍，身高，體重和 IQ 的分佈屬之，子彈在靶子上的分佈亦屬之。鐘形曲線的存在完全是由於許多小的，獨立誤差或隨機異變(Variation)來源總和所導致的結果。瞄準靶標時所出現的隨機誤差常態分佈可看出幾種因素－也許由於手的抖動，子彈的變異，槍身的不精確等等。身高的常態分佈顯示身高不是由單一遺傳因子所控制，而是由許多互為獨立的遺傳因子一起控制的。

多重獨立誤差總和導致鐘形曲線的事實提示我們將一些獨立亂數加起來就可以很容易地產生出該曲線：

```
10 R = INT (RND(1) + RND(2) + RND(3) + RND(4)
+ RND(5) + RND(6) +RND(7) + RND(8) + RND(9) +
RND(10))
```

or

```
10 FOR J = 1 TO 10: R = R + INT (RND (J))):
NEXT J
```

鐘峯的尖突程度依所加之RND數目的個數而定。當只有一個亂數時，整個情況就回到前面的X＝RND(1)去，而兩個數目相加就會向一個中間數值集中。如果把兩個介於1與6之間的整數相加，我們就相當於在擲一對骰子，它們相加之後的數目介於2與12之間。注意，並非所有的和都像前面的矩形分佈一樣有相等的出現概率。擲出2點的方法只有一種，那就是兩個1點；擲出3點的方法有兩種：1＋2或2＋1；擲出4點的方法有三種：1＋3，2＋2和3＋1；擲出5點的方法有四種：1＋4，4＋1，2＋3和3＋2；擲出6點的方法有五種（1＋5，5＋1，2＋4，4＋2和3＋3），而擲出7點的方法有六種（1＋6，6＋1，2＋5，5＋2，3＋4和4＋3）。對應於上述各種情況，擲出8，9，10，11或12的方法分別有5，4，3，2和1種。因此，在模擬投擲兩個以上骰子的亂數序列中，中間範圍的數目出現的機會將最爲頻繁，而大小兩邊的數目將比較少；下面的程式可以看出這個狀況。（其中的數目已調整爲介於0與9之間，不是1與12之間。它也可以模擬任意數目骰子的投擲，不僅僅是兩粒骰子。）爲了方便比較，這程式也產生一組各個獨立，介於0與9之間的隨機整數，重點是在這些數目具有相等之出現頻率，不像幾個骰子相加後的數

目則集中在中間數目。

本程式還將亂數按對乘在一起（130那一列），因爲每個亂數都是介於0與1之間的小數，若將它乘上一個數目，它就會變成該數目的大小程度，因此130那一列會產生一連串大多包含很小數目的亂數列；亦即，它被故意拉向小數目。

```
90 INPUT "HOW MANY 'DICE' THROWN FOR NORMAL
DISTRIBUTION ?  ";KMAX
100 FOR G = 1 TO 1000
110  REM CALCULATE RECTANGULAR, SKEWED AND
'NORMAL' RANDOM NUMBERS -----
120  R = INT (10 * RND (1)): REM RECTANGULAR
DISTRIBUTION
130  S = INT (10 * RND (1) * RND(1)): REM
SKEWED TOWARDS SMALL NUMBERS
140  FOR K= 1 TO KMAX
150    N = N + RND (1)
160  NEXT K
170  N = INT (N * 10/KMAX): REM NORMAL

DISTRIBUTION
180  REM ADD UP FREQUENCIES -----
190  RR(R) = RR(R) + 1
200  SS(S) = SS(S) + 1
210  NN(N) = NN(N) + 1
220 NEXT G
230 REM PRINT OUT FREQUENCIES -----
240 PRINT "#  R","S","N"
250 FOR J = 0 TO 9
260  PRINT J;"  ";RR(J), SS(J), NN(J)
270 NEXT J


RUN

HOW MANY 'DICE' THROWN FOR RANDOM
DISTRIBUTION ?  10

#     R        S        N

0    111      365        0
1    121      179        1
2     77      111       30
3    108      115      139
4     91      107      359
5    131       52      327
6     94       31       91
7     75       27       53
8     94       13        0
9     98        0        0
```

本程式印出一份包含以上三種頻率分佈的表。在矩形分佈R中，每一個0至9的數值J出現的次數大致相等一各有大約100 次，在取樣誤差的限度之內。在斜形分佈S中，零出現得最多，而數目愈大出現的次數愈少。在常態分佈N中，中間的數值最普遍，而愈往兩邊的數值則出現得愈來愈少。

若想獲得這些分佈的低解度漂亮圖形，可將230那一列改爲：

```
230 GR
240 FOR J = 0 TO 9
250 COLOR = 3: VLIN 39 - RR(J)/10, 39 AT J
260 COLOR = 6: VLIN 39 - SS(J)/10, 39 AT J +
15
270 COLOR = 9: VLIN 39 - NN(J)/10, 39 AT J +
30
280 NEXT J
```

低解析度圖形的擲骰子遊戲

要模擬投擲一對骰子所產生的亂數是很簡單的，您只需：

```
10 D1 = 1 + 6 * RND(1)
20 D2 = 1 + 6 * RND(2)
30 PRINT D1 + D2
```

但是如果您的眼睛能夠看到與聽到一對骰子在轉將會更刺激。下面就是以低解析度圖形（幾乎）完成這件事的Applesoft程式，您可以將它納入您自己的賭博遊戲中。

```
30 TEXT: HOME: GR
40 FOR THROW = 1 TO 5 + 10 * RND (3)
50 FOR J = 1 TO 2: D(J) = 1 + 6 * RND (1)
70   SOUND = PEEK (-16336) - PEEK (-16336)
80   X = 32
90   Y = 10 + 10 * J
100   COLOR= 1 + 15 * RND (1): REM FIELD
110   FOR H = X TO X + 6
120     VLIN Y, Y + 6 AT H
130   NEXT H
140   COLOR = 0 : REM DOTS
```

```
150   ON D(J) GOSUB 1100, 1200, 1300, 1400,
1500, 1600
160 NEXT J
170 NEXT THROW
180 VTAB 22: PRINT "***   HIT SPACE BAR FOR
NEXT THROW   ***"
190 GET K$
200 GOTO 10
210 END

1100 REM ONE
1110 PLOT X + 3, Y + 3
1120 RETURN
1200 REM TWO
1210 PLOT X + 1, Y + 1: PLOT X + 5, Y + 5
1220 RETURN
1300 REM THREE
1310 GOSUB 1100: GOSUB 1200
1320 RETURN
1400 REM FOUR
1410 GOSUB 1200
1420 PLOT X + 5, Y + 1: PLOT X + 1, Y + 5
1430 RETURN
1500 REM FIVE
1510 GOSUB 1400: GOSUB 1100
1520 RETURN
1600 REM SIX
1610 GOSUB 1400
1620 PLOT X + 1, Y + 3: PLOT X + 5, Y + 3
```

您可以看得出來 50 那一列撿取兩個介於 1 與 6 之間的亂數。 80 至 130 那幾列隨機挑選顏色畫出兩個低解析度方塊作爲骰子，而 140 至 160 那幾列藉呼叫 1100 至 1600 那幾列內的副程式在每個骰子中畫出 1 至 6 個黑點。骰子上點的顏色與數目快速地變化著，其變化次數介於 5 與 15 之間，完全由 40 那一列內 THROW 的數值決定。 70 那一列在每次骰子數值變化時插入一個卡塔聲。最後，骰子趨於靜止，點數也就可以讀出來了 。

骰子會出現在螢光幕的右下方（ 80 和 90 那兩列，X＝ 32 ， Y＝ 10 或 20）。如果您在遊戲中採用了這個副程式，您可以藉改變 X 和 Y 的數值將骰子移到別處。比方說，如果有兩位遊戲者，您可以把一位遊戲者的骰子放在左邊，另一位遊戲者的骰子放在右邊；如果有四位遊戲者，您可以把骰子放在螢光幕的四個角落。

這個副程式可在 Diddle（美式骰戲 2）與 Craps（美式骰戲 1）中找到。

在遊戲程式中加註標題

一旦您寫好了遊戲程式後最好在頂部加一段標題，這當然不會讓遊戲者在遊戲過程中看到，它只會在 LIST 程式時出現在一組 REM 敍述中。

標題的設計法因人而異，以下是筆者的做法：在附有自動啓動（Autostart）ROM 的 Apple Ⅱ 和 Apple Ⅱ Plus 上利用 ESC－I， ESC－M，ESC－J 和 ESC－K 四個編輯鍵。（如果您的 Apple Ⅱ 沒有自動啓動 ROM，則您不會有上述四個編輯鍵，然而您可以利用 ESC － A ， ESC － B ， ESC －C 和 ESC － D－或者乾脆跳過本節不看。）讓我們假定您已經很熟悉這些編輯鍵的用法。

首先，在一張草稿紙上大致描出您想要的標題。例如：

```
10 REM * * * * * * * * * * * * * * *
20 REM *                           *
30 REM *   TIC TAC TOE             *
40 REM *                           *
50 REM *      BY                   *
60 REM *                           *
70 REM * STUART ANSTIS             *
80 REM *                           *
90 REM * * * * * * * * * * * * * * *
```

這個標題是以星號將遊戲的名稱和作者姓名框起來。注意， 30 ， 50 和 70 那幾列包含遊戲名稱，“ by ”和作者姓名； 20 ， 40 ， 60 和 80 那幾列除了頭尾各一個星號外其餘均爲空白； 10 和 90 那兩列各以一列星號作爲頂線和底線。

您也可以只打出標題那幾個字就算了，但利用編輯鍵可以省去許多打字工作。首先，選出最長的一列一遊戲名稱或作者姓名；讓我們假定它是 70 那一列的作者姓名。假設我們要打：

```
70REM*  STUART ANSTIS  *
```

先打入 70REM *（中間不留空白），再打兩個空白，接著打作者姓名，再打兩個空白，最後打一個星號再按下RETURN鍵。這樣就可以定義出每一列所需的長度，以便確保最重要的一件事一將所有星號對齊。

利用編輯鍵將游標移到這一列的第一個字元“ 7 ”上。

```
70REM*  STUART ANSTIS  *
```

隨便敲一個鍵離開編輯模式，再按 5 將 70 改爲 50 。利用 ESC－K 令游標派司過最前面幾個字元，停在第六個字元上。

```
50REM*  STUART ANSTIS  *
```

然後利用空白鍵擦去“ STUART ”，打入“ BY ”，再利用空白鍵擦掉作者姓名的其餘部分，一直到第二個星號前面

```
50REM*      BY        *
```

然後利用 ESC－K令游標派司過該星號，再按RETURN 。暫時還不要 LIST 任何東西，因爲 LIST 命令會在輸出上填塞一些空白，這是我們還不想要的。

現在，再把游標移到 50 那一列的第一個字元“ 5 ”上，將它重新打成

```
30REM*   TIC TAC TOE  *
```

接下去，做 20 ， 40 ， 60 和 80 那幾列；反覆將游標置於第一個字元上，再利用空白鍵擦去星號之間的一整列字元。

最後，回到 80 那一列將它改爲 10，並填滿星號。做好後將 10 再改爲 90 ，並複製一列。

```
LIST

10 REM *******************
20 REM *                 *
30 REM *   TIC TAC TOE   *
40 REM *                 *
50 REM *       BY        *
60 REM *                 *
70 REM * STUART ANSTIS   *
80 REM *                 *
90 REM *******************
```

注意一點：如果您利用圖形來做快速動作遊戲，您應該將繪圖副程式放在程式最前面，甚至應該放在標題之前。這是因爲電腦並不十分聰明，雖然命令 GOSUB 500，它並不是直接跳到 500 那一列，而是從程式的開頭向下找，一直到找到 500 爲止。因此，在這種情況下可將副程式置於 10 那一列處，您甚至可以取消標題。

徵求朋友的意見

好了，假設您已經完成一個遊戲程式，而且反覆檢查過許多遍，已經除去所有的錯誤，並保證不會讓人看不懂，您很可以在朋友面前展示一番。您認爲您的工作完了嗎？您會大吃一驚的，請咬牙準備接受批評吧！

請一些朋友來，讓他們坐在鍵盤前面，再在您的膝上放一本筆記簿，然後 RUN 那遊戲程式。現在，坐回去，注意看！讓我們假設螢光幕清除之後顯示出：

```
WANT INSTRUCTIONS (Y/N) ?
```

您以爲它的意義很清楚，打“ Y ”代表是的，打“ N ”代表不要，不過，您的朋友怎麼樣呢？如果他是玩電腦的，他大概想都不會想一下就會打 Y 或 N，就像您一樣。如果他不是，他很可能大聲地一個字一個字讀出來，然後轉而問您那是什麼意思。別告訴他，什麼也別講！讓他自己解決問題。記住，如果您的遊戲

要在雜誌上發表，或提供給玩家俱樂部，或者成爲商品，您不可能跟着到處去解釋它，因此必須讓遊戲自己來做這件事。

如果您的朋友眞的不知該怎麼辦，您可在筆記簿上記下來；很明顯地，您還需要另一層說明。事後您可在程式中另加一列：

```
PRINT "HIT 'Y' FOR YES, 'N' FOR NO."
```

如果他眞的了解上面的說明，他是打 Y 呢，還是 YES 呢？他按了 RETURN 鍵嗎？不管那種情形，您的程式有一列 GET Y$ 或 INPUT Y$ 在裡面嗎？對於後者，您的程式會徘徊至他記得按下 RETURN 爲止；然而對於前者，如果他意外地按了 RETURN 會怎樣？這種情況可能會觸發程式中接下去的另一個 GET 敍述而導致意想不到的結果。您考慮過這些問題嗎？

這一點看起來很令人沮喪，但這種問題的確會發生。通常您不應該看到遊戲者坐在電腦前面等待電腦做什麼事，而是應該讓電腦等遊戲者做什麼事。您可以觀察遊戲者的反應，並將結果記下來以改善所看到的問題；您也可以從這上面學到不少東西。

朋友可以回饋您兩種情報，如果情報沒什麼用處，您可以從他們的錯誤中知道如何使程式不會讓人不知所措。如果他們的建議很有用，您可以從他們建設性的指正中學到如何使程式更有趣，更能娛樂遊戲者。先把他們當作白痴，再把他們當做聰明人，請他們提供建議。他們的建議可能很平常，但也可能很有用一例如“我比較喜歡鍵盤，比較不喜歡按鈕”或者“您要我用左手控制‘1’鍵，用右手控制‘0’鍵來控制左右，但是我比較喜歡用‘1’和‘2’鍵，因爲我只要用一隻手就夠了。”您的朋友還可能有其他更一般性的想法：“這種太空爭霸戰（space war）是單方向的，您能讓它有時候從背後攻擊我嗎？”或者“可

以讓我選擇一個人玩或兩個人玩嗎，這樣我可以跟我自己比賽，或者跟朋友一起玩。”

一個好遊戲應該怎樣？

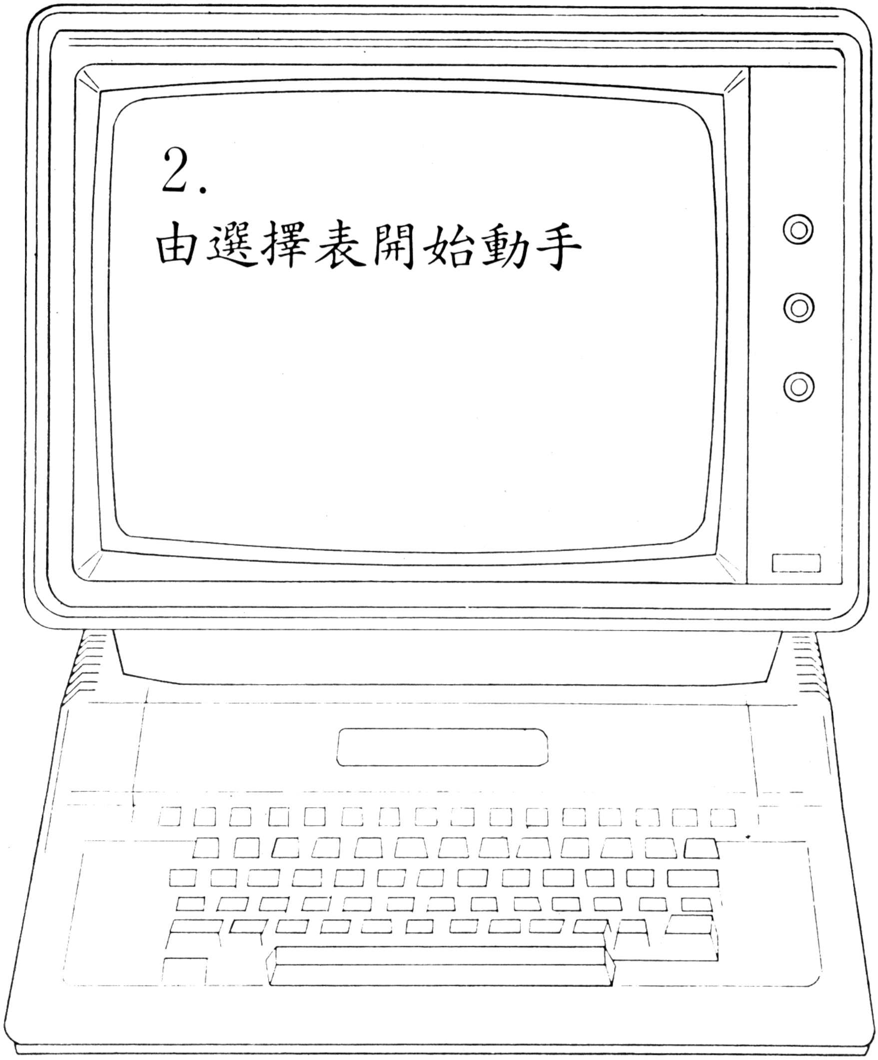
一個好的遊戲應該像一架鋼琴一樣：很容易彈得不好，又很不容易彈得很好。它必須很容易開始，以便讓生手有興趣繼續玩，而不會畏縮不前，而且應該有足夠的挑戰性，讓熟練的人保持興趣。有一個達到這目標的方法就是將遊戲分成難易數個等級。例如，在圖形動作遊戲中可讓遊戲者分數的高低與移動靶標的速度連在一起；生手的分數較低，因此令靶標移動得慢一點，以使他產生興趣，高手的分數較高，可讓他對抗非常快速，較難擊中的目標。

一般商業遊戲製造商通常把遊戲分成主題與機構兩部分，他們認爲機構比較重要，認爲它是遊戲的核心，而主題只是裝飾用，可以隨意更換。例如，市面上有許多電腦化的探險遊戲，它們的機構是相同的，只是主題不同而已一有的設在外太空，有的在地下洞穴裡，而有的在妖精國裡，但遊戲的結構是大同小異的。到電動遊樂場去看看主題與機構的差異是值得的，那種遊戲都是那一行中最佳的職業好手在强大的市場壓力下寫出來的。您可以注意看看那一台機器比較吸引觀衆，仔細研究那些遊戲，您將會看到它們幾乎都提供難易程度不同的等級，讓生手開始時只得幾百分，但經過幾個月的練習之後（一次 10 塊錢！）就可進入高手行列。有許多遊戲記錄最高的得分者，如果您在某部機器上打破前面留下的記錄，您可以把名字打入機器，電腦就會向後來的遊戲者顯示您的大名與分數一直到關掉機器，或有其他人打破您的記錄。這種方式的確能夠引誘一些遊戲者一直將錢投進去，您可以將這種功能納入您的遊戲程式中，只要利用一個副程式將最

佳的前十名分數儲存在磁碟片上，並在遊戲時顯示出來即可。

不過，您應該知道商業電動遊戲都是以非常快速的機器語言寫成的，因此您不能期望您天生就慢的 BASIC 語言描繪出那種閃電般快速的圖形。如果您想要速度快，您必須利用形狀表，商業套裝繪圖程式或 Apple 機器語言。

現在，您應該已經充分掌握住寫作 BASIC 遊戲的原理了，讓我們看一些遊戲，您可以在裡面找找本章所說明的方法和副程式，看看它們是如何納入程式中的。利用鍵盤打入遊戲，將它們 SAVE 在磁碟片上，並試著玩玩看。您可以自由添加點東西或改良它們——以它們作爲創作的跳板。祝您成功！



"Hello"程式

每當您啟用一張新磁碟片時您最好都放一個在您 boot （啟動）磁碟片時電腦會自動執行它的 "Hello" 程式。它通常是一個很簡單的程式，例如：

```
10 PRINT "MASTER DISKETTE, 48K"
20 PRINT "JOHN SMITH, 1983"
30 END

RUN

MASTER DISKETTE, 48K
JOHN SMITH, 1983
```

但是，它事實上可以是任何一個您希望的程式。本章的〝Hello〞程式是一個可將Apple變成一部極易操作（turnkey）系統的選擇表（menu），這是說使用者根本無需了解電腦是什麼就可以起動並操作它，就好像轉鑰匙一樣容易。

選擇表程式使得從磁碟片上選擇程式容易得好像在餐館內選菜一樣。如果您想要一客在餐館菜單上列在第7項的蘋果派，您只要向侍者叫第7號即可。大多數餐館爲了方便起見都將食物區分成湯，主菜，甜點等類別。在這個選擇表程式中，遊戲也被區分成Action Games（動作遊戲一），More Action Games（動作遊戲二），Strategy Games（策略遊戲），……等等。

這個程式的使用本身就很簡單（這也正是利用選擇表的原意），當您boot磁碟片時下面顯示會出現在螢光幕上：

```
******************************************
              MENU

******************************************

WOULD YOU LIKE:

1. ACTION GAMES
2. MORE ACTION GAMES
3. BOARD GAMES OF STRATEGY
4. WORD GAMES
5. EDUCATIONAL/SCIENTIFIC
6. GRAPHICS
7. MORE GRAPHICS
8. SOLO GAMES
9. CATALOG

HIT 1, 2, 3, 4, 5, 6, 7, 8 OR 9

HIT ANY OTHER KEY TO RETURN TO MENU

DO NOT HIT RETURN
```

假設您只有自己一個人，因此您決定試試一個單人遊戲（Solo game）。只要在鍵盤最上面一排敲下〝8〞，螢光幕的顯示會像變魔術似的變爲：

```
****************************************
              SOLO GAMES

****************************************
WOULD YOU LIKE:

1. MEMORY MAN
2. ORGAN
3. MUSIC
4. ELITHORN
5. RUNNING ON EMPTY
6. BURIED TREASURE
7. GUESS A NUMBER
8. JUMPERS
9. PEGGY

HIT 1, 2, 3, 4, 5, 6, 7, 8 OR 9

HIT ANY OTHER KEY TO RETURN TO MENU

DO NOT HIT RETURN
```

若想玩 Elithorn（迷宮），可按〞4〞，程式便會自動從磁碟片載入（L-OAD）它，並自動 RUN，您無需做任何動作。選擇表具有〞樹〞的結構——剛開始時您選擇遊戲類別（動作遊戲，策略遊戲，…等），然後由程式送您到正確的分支去，您接下去選擇一個遊戲，程式再送您到所選擇的遊戲——〞決策樹〞的〞末梢〞去。

當您完成 Elithorn 遊戲時有一段提示訊息會出現在螢光幕上：

```
WANT ANOTHER GAME (Y/N) ?
```

如果您打入〞Y〞，遊戲會再重頭開始；如果您打入〞N〞，提示訊息會問您要不要回到選擇表去選擇其他遊戲：

```
WANT TO RETURN TO MENU (Y/N) ?
```

如果您打〞Y〞，選擇表程式將會重新出現。

但是，假定您剛剛才將單人遊戲（Solo Games）名單列在螢光幕上，正好門鈴響了，有一位朋友來找您比賽動作遊戲，您怎麼辦？很簡單，隨便敲一個鍵

——空白鍵最方便——單人遊戲名單就會消失，而主選擇表會重新出現，您就可以按〃1〃去選擇動作遊戲一（Action Games）。無論什麼時候只要您按了空白鍵，程式便會自動回到主選擇表去。這種安全網功能常被稱爲〃系統設定之情況〃（default)一當您不知道下一步該做什麼時都可以隨便按一個鍵，讓電腦去傷腦筋。

順便一提的是主選擇表上的

```
9. CATALOG
```

如果您在此時打入〃9〃，目錄（Catalog)將會出現在螢光幕上；亦即，磁碟片上所有程式的名字都會出現在螢光幕上。若想在目錄顯示之後回到選擇表程式內，可打

```
RUN HELLO
```

然後按RETURN鍵。

選擇表程式是一個處理字串的練習，所有遊戲名稱都被放入一個字串陣列（string array)中，然後PRINT 在螢光幕上，一次六個。遊戲者藉打入對應的號碼來選擇遊戲，程式便會RUN該遊戲。

160 那一列呼叫 740 至 1380 那幾列中包含遊戲名稱的副程式，每一個遊戲的名稱都是以字串形式GAME $（A，B）存在的。事實上，本程式中的GAME $字串有兩類，大部分是遊戲的名稱（當A 大於或等於 1 時），但有一些是遊戲的類別（當A＝0）。請看 750 至 770 那幾列：

```
750 GAME$(0,1) = "ACTION GAMES"
760 GAME$(0,2) = "MORE ACTION GAMES"
770 GAME$(0,3) = "BOARD GAMES OF STRATEGY"
```

這些是顯示出來供遊戲者選擇的遊戲類別。如果您打〃3〃，程式顯示的是各個策略遊戲的名稱，亦即：

```
980 GAME$ (3, 1) = "STOCK MARKET"
990 GAME$ (3, 2) = "TICTACTOE"
....
1040 GAME$ (3, 7) = "ZIGZAG"
```

如果您再打〃2〃，程式將執行GAME $（3，2）—Tic - Tac - Toe （井字遊戲）。

本程式200至310那幾列內的主迴圈連續呼叫三個主要副程式：♯1副程式（360那一列）將那六個一組遊戲（或遊戲類別）從遊戲者選擇的項目中PRINT在螢光幕上。♯2副程式（660那一列）PRINT出遊戲說明，例如，如果有三個遊戲可供選擇，則顯示〃HIT1，2OR3〃，或者如果有六個遊戲可供選擇，則顯示〃HIT 1，2，3，4，5 OR 6〃。爲了要知道必須顯示出多少數字，這副程式又呼叫一個位於1400那一列，可以計算顯示在螢光幕上遊戲數目的副程式。♯3副程式（500那一列）讀取遊戲者所按之數目HIT $（250那一列爲：GET HIT $）。假設主選擇表已顯示出來，提供了 "ACTION GAMES" (GAME$ (0, 1))至 "CATALOG" (GAME$ (0, 9)) 諸類選擇。如果遊戲者打入一個諸如4的數目，螢光幕之顯示會變爲猜字遊戲那一組：〃HANGMAN〃（GAME $ （4，1））至〃ANAGRAMS〃（GAME $ （4，6））。如果他又再打入4，他將會選到GAME $（4，4）的〃BIRTHDAY CARD〃（生日卡）遊戲。410那一列將會從磁碟片上LOAD這個遊戲，並RUN它：

```
410 IF A > 0 AND HIT > 0 THEN PRINT
CHR$(4);"RUN";GAME$ (A, HIT)
```

"IF A > 0 AND HIT > 0" 保証選到的是遊戲，不是遊戲的目錄（代表目錄的變數是GAME $（0，#）；亦即A＝0）。PRINT CHR$(4);"RUN..." 是一個磁碟作業系統（DOS）指令；CHR $(4)等於Control－D，它告訴DOS去執行引號內的命令。詳情請參閱DOS使用手册。

這是一個很有彈性而又通用的程式，您可以加以修改作爲己用，您也可以隨意增加一些遊戲。假設您寫了一個稱爲Master Brain的新猜字遊戲，並且將它存放在磁碟片上。因爲它是一個猜字遊戲（word game），所以它屬於第4類（780那一列：GAME $（0，4）＝"WORD GAMES"）。選擇表上已經有六個猜字遊戲（1100那一列：GAME $（4，6）＝"ANAGRAMS"），因此Master Brain將成爲GAME $（4，7），只要打入：

```
1105 GAME$ (4, 7) = "MASTER BRAIN"
SAVE HELLO
```

就成了，其他事情電腦自會處理。Master Brain將會出現在猜字遊戲的名單中，並在遊戲者選擇它的時候自動執行。爲了安全起見最好將Hello保存（SAVE）在一張備份（backup）磁碟片上，以防意外事件發生把資料損毀了，這樣您就不會失去原來辛辛苦苦打成的Hello程式；磁碟片是很容易在意外情況下出錯的。

這種選擇表程式的用處不止如此，您可以把它應用在任何包含一組磁碟程式的地方，不一定非得遊戲程式不可。

若想把它用在您自己的磁碟片上，您首先應該删去現有包含遊戲名稱的副程式：

```
DEL 740, 1330
```

再打入您自己的程式分類表，例如：

```
740 GAME$ (0, 1) = "UTILITY PROGRAMS"
750 GAME$ (0, 2) = "HI-RES GRAPHICS"
760 GAME$ (0, 3) = "STATISTICS PROGRAMS"
```

然後加入每一類的各個程式，例如：

```
1000 GAME$(1,1) = "CHECK DISK SECTORS"
1010 GAME$(1,2) = "HEX-DECIMAL CONVERSION"
1020 GAME$(2,1) = "DRAW A SPIRAL"
1030 GAME$(2,2) = "HI-RES COLOR DEMO"
1040 GAME$(2,3) = "PLOT A SQUARE"
1050 GAME$(3,1) = "LINEAR REGRESSION"
```

最後，將您的選擇表程式存在磁碟片上：

```
SAVE HELLO
```

從今以後，每當您 boot 該磁碟片時這個選擇表（Menu）程式都會自動出現，您會發現這種方法比以前要叫出目錄，並回憶每個程式做什麼用的老方法容易多了。

```
10  REM  *******************
20  REM  *                 *
30  REM  *      HELLO      *
40  REM  *                 *
50  REM  *     (MENU)      *
60  REM  *                 *
70  REM  *       BY        *
80  REM  *                 *
90  REM  *  STUART ANSTIS  *
100  REM *                 *
110  REM *******************
120  REM
130  REM
140  DIM GAME$(10,20),W(20)
150  TEXT : HOME
160  GOSUB 740: REM  NAMES OF GAMES
170 A = 0:HIT = 0
180  GOSUB 340
190  REM
200  REM *********************
210  REM          MAIN LOOP
220  REM *********************
230  REM
240  GOSUB 630: REM  "HIT 1234" INSTRUCTIONS
250  GET HIT$
260  GOSUB 510: REM  READ KEYPRESS VALUE
270  GOSUB 340: REM  PRINT FRAME OF TEXT
280 A = HIT
290  GOTO 240
300  REM
310  REM *********************
320  REM  END OF MAIN LOOP
330  REM *********************
340  REM  PRINT FRAME OF TEXT -----
350  HOME : VTAB 2
360  PRINT "****************************************"
370  PRINT
380  HTAB 14: PRINT GAME$(A,HIT): PRINT
390  PRINT "****************************************"
400  PRINT
410  IF A > 0 AND HIT > 0 THEN  PRINT  CHR$ (4);"RUN ";GAME$(A,HIT)
420  IF A = 0 AND HIT = 9 THEN  PRINT  CHR$ (4);"CATALOG": END
430  PRINT "WOULD YOU LIKE:"
440  PRINT
450  GOSUB 1390
460  FOR J = 1 TO W
470  PRINT J;"  :";GAME$(HIT,J)
480  NEXT J
490  RETURN
500  REM  READ KEYPRESS VALUE -----
510 HIT =  VAL (HIT$)
520  IF HIT < W + 1 THEN  RETURN
530  GOSUB 550
540  RETURN
550  REM  INCORRECT KEY PRESS -----
560  HOME : VTAB 5
570  PRINT "YOU MUST HAVE HIT A LETTER"
580  CALL  - 1059
590  FOR T = 1 TO 500: NEXT T
600 HIT$ = "0"
610  RETURN
620  REM   "HIT 1,2,3,4..." INSTRUCTIONS -----
630  PRINT : PRINT "HIT ";
640  FOR J = 1 TO W - 1
650  PRINT J;",  ";
660  NEXT J
670  PRINT "OR ";W
680  PRINT
690  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
700  PRINT
710  PRINT "DO NOT HIT RETURN"
720  RETURN
```

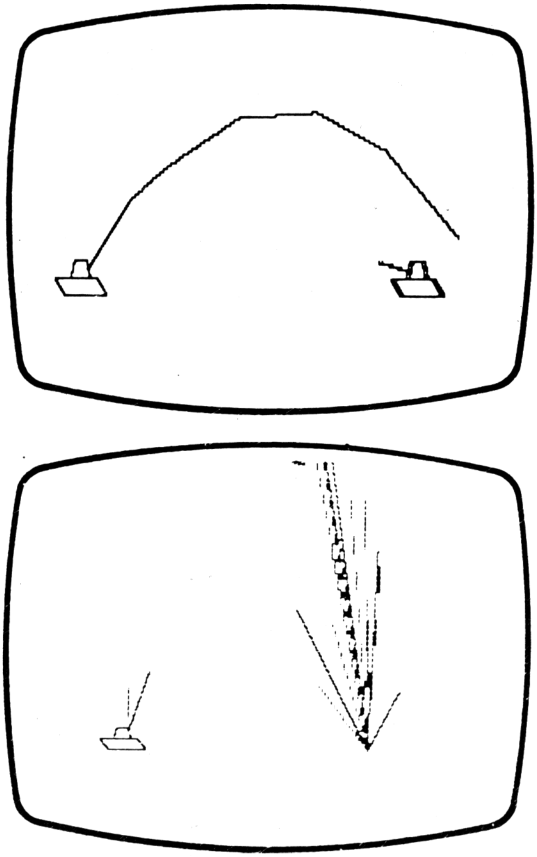
```
730  REM  NAMES OF GAMES -----
740 GAME$(0,0) = "MENU"
750 GAME$(0,1) = "ACTION GAMES"
760 GAME$(0,2) = "MORE ACTION GAMES"
770 GAME$(0,3) = "BOARD GAMES OF STRATEGY"
780 GAME$(0,4) = "WORD GAMES"
790 GAME$(0,5) = "EDUCATIONAL/SCIENTIFIC"
800 GAME$(0,6) = "GRAPHICS"
810 GAME$(0,7) = "MORE GRAPHICS"
820 GAME$(0,8) = "SOLO GAMES"
830 GAME$(0,9) = "CATALOG"
840 GAME$(1,1) = "TANK"
850 GAME$(1,2) = "SPRINT"
860 GAME$(1,3) = "BOMBER"
870 GAME$(1,4) = "WAIT FOR IT"
880 GAME$(1,5) = "HORSE RACE"
890 GAME$(1,6) = "WEIGHT LIFTERS"
900 GAME$(2,1) = "TARGET"
910 GAME$(2,2) = "HELICOPTER"
920 GAME$(2,3) = "TIDDLER"
930 GAME$(2,4) = "PING PONG"
940 GAME$(2,5) = "DIDDLE"
950 GAME$(2,6) = "SPACE ZAPPER"
960 GAME$(2,7) = "CRAPS"
970 GAME$(2,8) = "CROSS PORPOISES"
980 GAME$(3,1) = "STOCK MARKET"
990 GAME$(3,2) = "TICTACTOE"
1000 GAME$(3,3) = "TURNOVER"
1010 GAME$(3,4) = "CHOMP"
1020 GAME$(3,5) = "GOLD AUCTION"
1030 GAME$(3,6) = "BRIDGE-IT"
1040 GAME$(3,7) = "ZIGZAG"
1050 GAME$(4,1) = "HANGMAN"
1060 GAME$(4,2) = "WHAT'S YOUR NAME?"
1070 GAME$(4,3) = "ANY MESSAGES?"
1080 GAME$(4,4) = "BIRTHDAY CARD"
1090 GAME$(4,5) = "WHAT'S MY WORD?"
1100 GAME$(4,6) = "ANAGRAMS"
1110 GAME$(5,1) = "MAGIC SQUARES AND CUBES"
1120 GAME$(5,2) = "ARROW"
1130 GAME$(5,3) = "RANDOM WALK"
1140 GAME$(5,4) = "SCATTER"
1150 GAME$(5,5) = "PASCAL'S TRIANGLE"
1160 GAME$(5,6) = "ANIMATED SKETCHPAD"
1170 GAME$(6,1) = "BUTTERFLY"
1180 GAME$(6,2) = "FIREWORK"
1190 GAME$(6,3) = "ELECTRIC LORES"
1200 GAME$(6,4) = "ELECTRIC HIRES"
1210 GAME$(6,5) = "POLYGONS"
1220 GAME$(6,6) = "SQUIRALS"
1230 GAME$(7,1) = "LISSAJOU"
1240 GAME$(7,2) = "STARS"
1250 GAME$(7,3) = "DANDELIONS"
1260 GAME$(7,4) = "DOILY"
1270 GAME$(7,5) = "STOPWATCH"
1280 GAME$(8,1) = "MEMORY MAN"
1290 GAME$(8,2) = "ORGAN"
1300 GAME$(8,3) = "MUSIC"
1310 GAME$(8,4) = "ELITHORN"
1320 GAME$(8,5) = "RUNNING ON EMPTY"
1330 GAME$(8,6) = "BURIED TREASURE"
1340 GAME$(8,7) = "GUESS A NUMBER"
1350 GAME$(8,8) = "JUMPERS"
1360 GAME$(8,9) = "PEGGY"
1370 GAME$(9,1) = "CATALOG"
1380  RETURN
1390  REM  COUNT NUMBER OF GAMES IN ARRAY -----
1400  PRINT
1410 W = 1
1420  IF GAME$(HIT,W) <  > "" THEN W = W + 1: GOTO 1420
1430 W = W - 1
1440  RETURN
```



坦克大決戰(Tank)

在這個高解析度圖形動作遊戲中，您要指揮螢光幕左下方的坦克車射擊由右方接近攻擊您的任何敵方坦克。您必須利用 1 號遊戲控制器藉調整您這一方坦克礮管的仰角來設定射程，並利用控制器按鈕發射砲彈。每一發砲彈。每一發砲彈都會向敵方坦克產生一條拋物線彈道。如果您沒有擊中，敵方坦克會一直朝您進逼，企圖壓過您的位置將您剷除；如果您擊中該坦克，則它會爆炸，然後消失掉。但隔一會兒又會有一輛新坦克發動新攻勢，而且速度會比前一輛更快，所以您必

須保持高度警覺。



把敵火瞄準進犯之坦克

這個遊戲包含一組相互關連的高解析度圖形副程式。貫穿整個遊戲的是一段描繪坦克車的副程式，因此我們將它放在最前面，甚至放在程式標題之前，以確保獲得最快執行速度。記住，Applesoft 無法直接跳到副程式所在的列號上，它必須在程式中一列一列地找。

125 至 320 那幾列設定變數的初值，將 TGT 設定爲 6 ，這是您這一方坦克的位置，再呼叫 30 那一列的副程式去描繪您的坦克，並將它留在那兒。300 至 320 那幾列從某個隨機位置 TGT 起動敵方的坦克，並呼叫 30 那一列的副程式去描繪敵方坦克（ HCOLOR＝7 ），接著擦掉它（ HCOLOR＝0 ）然後向前移

動一點（ TGT＝TGT－PACE ），再重新描繪它。同時，您可以利用1號控制器在322 －－440 那幾列中瞄準敵方坦克；若要發射砲彈可按下您的控制器按鈕（PEEK(－16286)），它便會呼叫490 至680 那幾列內的射擊副程式。這個副程式定義砲彈的拋物線飛行彈道方程式：

```
X = COS (A) * T
Y = SIN (A) * T - G * T * T / 2
```

爲了加速程式的執行，正弦值MS與餘弦值MC在1370至 1430 那幾列中已預先算好。X與Y的數值必須再乘上砲彈離開礮口的速度MUZZLE（其值爲3000）。此外，T是時間，G是重力加速度，而A是礮管的仰角，只要改變礮管仰角就可改變砲彈的射程。仰角爲45 度時可獲得最大射程，仰角愈大射程愈短；（如果您把礮管向上垂直會怎樣？）在本遊戲中，仰角是無法低於 45 度的。

您可曾擊中敵方坦克？在1450至1520 那幾列中，當砲彈擊中地面時（ Y＝159 ），如果彈坑距離礮塔24 個單位以內（X＞TGT－1 AND X＜TGT＋24），您將被記錄擊中一次，程式便會GO TO 690，令敵方坦克爆炸，然後消失；爆炸副程式係抄自高解析度示範程式Firework（烟火）。接著，敵方坦克的速度PACE會加2讓下一次決戰更形困難，然後程式回頭開始下一波攻擊。

Tank

```
10  REM  TANK22-----
20  GOTO 130
30  REM DRAW TANKS-----
40 HC = HUE: HCOLOR= HC
60 YY = 159
70  HPLOT TGT,YY TO TGT - 5,YY - T6 TO TGT + 18,YY - T6 TO TGT + 24,YY TO
     TGT,YY
80  HPLOT TGT + T4,YY - T6 TO TGT + T6,YY - 12 TO TGT + 13,YY - 12 TO TGT
      + 15,YY - T6
90  IF TGT > 12 THEN  HPLOT TGT + T4,YY - T9 TO TGT - 12,YY - 12
100  IF HUE = 0 THEN TGT = TGT - PACE
110  IF TGT > 6 AND TGT < 25 THEN OVER = 1: REM  FLAG FOR OVERRUN
120  RETURN
```

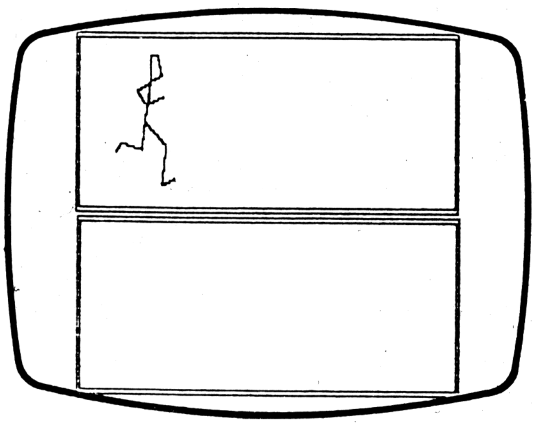
continued . . .

```
125  REM  INSTRUCTIONS-----
130  TEXT : HOME
140  PRINT "WANT INSTRUCTIONS (Y/N) ?"
150  GET Y$
160  IF Y$ = "Y" THEN  GOSUB 1220: REM  INSTRUCTIONS
170 S = 20: REM STEP SIZE FOR PARABOLA
180  DIM R(50),S(50),GT(30),T(30)
190 G = 32.2: REM GRAVITY
200  GOSUB 1370: REM  PARABOLA CALCULATIONS
220 PACE = 2
230 WAVE = 1
240 T2 = 2:T4 = 4:T6 = 6:T9 = 9: REM  FOR PLOTTING TANK
250  HGR
260 MUZZLE = 3: REM  MUZZLE VELOCITY
270 SHOT = 0
280 HUE = 7
290 TGT = 6: GOSUB 30: REM  DRAW OWN TANK
300 TGT = 150 + 100 *  RND (5)
310  FOR HUE = 0 TO 7 STEP 7: GOSUB 30: REM  DRAW ENEMY TANK
320  NEXT HUE
322  REM AIM GUN ----------
324  IF OVER = 1 THEN OVER = 0: GOTO 990: REM  OVERRUN
326  IF BANG = 1 THEN BANG = 0: GOTO 800: REM  EXPLOSION
328  IF HIT = 1 THEN HIT = 0: GOTO 690: REM  HIT
340 X = 19:Y = 150
350 R = 25
360 HOLD = H:VOLD = V
370  HCOLOR= 7
380 H =  INT ( PDL (1) / 10)
390 V =  INT ( SQR (R * R - H * H)) + H
400  HPLOT X,Y TO X + H,Y - V
410 HUE = HUE - 7
420  IF  PEEK ( - 16286) > 127 THEN  GOTO 490: REM  BUTTON PUSH
430  IF  ABS (H - HOLD) > 0.5 OR  ABS (V - VOLD) > 0.5 THEN  GOSUB 460: REM
      ERASE OLD GUN?
440  GOTO 310: REM DRAW ENEMY TANK &RE-AIM
450  REM  ERASE OLD GUN-----
460  HCOLOR= 0
470  HPLOT X,Y TO X + HOLD,Y - VOLD
480  RETURN
490  REM FIRE SHOT ON BUTTON PUSH-----
500 A =  ATN (V / (H + 0.1))
510 MS = MUZZLE *  SIN (A)
520 MC = MUZZLE *  COS (A)
540  GOSUB 940: REM   SOUNDS
550 SHOT = SHOT + 1
560  FOR FIRE = 5 TO 0 STEP  - 5
570  HCOLOR= FIRE
580  HPLOT 20 + H,150 - V
590  FOR J = 1 TO 25
600 X = MC * T(J) + 20: REM FORMULA FOR PARABOLIC FLIGHT
610 Y = 150 - MS * T(J) + GT(J)
620  IF X > 279 OR Y > 150 THEN  GOTO 1450
640  HPLOT  TO X,Y
650  NEXT J
660  NEXT FIRE
670  IF SHOT > 6 THEN  GOTO 1110: REM  OUT OF AMMO
680  GOTO 410
690  REM  SCORE A HIT ----------
700  HOME : VTAB 21
710  PRINT "YOU KNOCKED OUT TANK # ";WAVE;" !!!"
720  PRINT "IT TOOK YOU ";SHOT - 2;" SHOTS."
730  FOR W = 1 TO 1000: NEXT W
740 PACE = PACE + 2
760 WAVE = WAVE + 1
765 AMMO = AMMO + SHOT - 1
770  HOME
775 HIT = 0
780  GOTO 250
790  END
800  REM EXPLOSION ------
810  HCOLOR= 0: GOSUB 60: REM  ERASE TANK
820 YY = 159
830  HPLOT TGT,YY
840  FOR J = 15 TO 30
850  GOSUB 940
860 HC =  INT (7 *  RND (3)) + 1
870  HCOLOR= HC
880  IF R(J) > TGT THEN R(J) = TGT
890  IF S(J) > YY THEN S(J) = YY
```

待續……

```
900   HPLOT TGT + 10,YY TO TGT - R(J),YY - S(J)
910   NEXT J
915   IF FINISH = 1 THEN FINISH = 0: GOTO 1040
920   IF TGT < 25 THEN  GOTO 730
930   GOTO 700
940   REM : SOUNDS ----------
950   FOR M = 1 TO 2
960 SOUND =  PEEK ( - 16336) -  PEEK ( - 16336)
970   NEXT M
980   RETURN
990   REM    OVERRUN ----------
1000   VTAB 23
1010   PRINT "YOU HAVE BEEN OVERRUN BY THE ENEMY"
1030   PRINT "AFTER KNOCKING OUT "WAVE - 1;" ENEMY TANKS, USING UP ";AMMO;
   " SHOTS."
1035 FINISH = 1: GOTO 820: REM EXPLOSION
1040   INPUT "WANT ANOTHER GAME (Y/N) ";Y$
1050   IF Y$ = "Y" THEN  HOME :AMMO = 0: GOTO 220
1055   TEXT : HOME : VTAB 10
1060   PRINT "HIT 'ESC' TO ESCAPE"
1065   PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
1070   GET R$
1075   IF R$ =  CHR$ (27) THEN  TEXT : HOME : VTAB 10: PRINT "BYE.  THANKS
   FOR PLAYING."
1080   PRINT  CHR$ (13) +  CHR$ (4);"RUN HELLO"
1090   HOME
1170   REM : SHOOT YOURSELF ----------
1180   HOME : VTAB 21
1190   PRINT "YOU JUST SHOT YOURSELF.
1200   PRINT "YOU NOW HAVE A HOLE IN THE TOP OF YOUR": PRINT "HEAD."
1205   FOR W = 0 TO 800: NEXT W
1208   HOME
1210   GOTO 990: REM  OVERRRUN
1220   REM : INSTRUCTIONS ----------
1230   VTAB 6
1240   PRINT "YOUR TANK IS ON THE LEFT.  ENEMY TANKS"
1250   PRINT "ADVANCE FROM THE RIGHT.  TURN PADDLE #1"
1260   PRINT "TO AIM YOUR GUN.  45 DEG GIVES MAXIMUM"
1270   PRINT "RANGE: HIGHER ELEVATION GIVES SHORTER"
1280   PRINT "RANGE.  PRESS PADDLE BUTTON TO FIRE GUN."
1290   PRINT : PRINT "YOU MUST ALTER PADDLE BEFORE FIRING EACH SHOT."
1300   PRINT : PRINT : PRINT "    ***   GOOD LUCK !!! ***"
1310   PRINT : PRINT
1320   INVERSE
1330   PRINT " HIT ANY KEY TO BEGIN ...   "
1340   GET H$
1350   NORMAL : HOME
1360   RETURN
1370   REM  PARABOLA CALCULATIONS-----
1380   FOR J = 0 TO 30
1390 GT(J) = G * J * J / (S / 5): REM  WAS S*1.5 *****
1400 T(J) = S * J
1410 R(J) = 3 * J *  RND (J) - 1.5 * J
1420 S(J) = 10 * J *  RND (J)
1430   NEXT J
1440   RETURN
1450   REM  WHEN SHOT LANDS (Y=159)---
1460   IF X < 25 THEN  GOTO 1170: REM  SHOOT YOURSELF
1470   IF X > 279 THEN X = 279
1480   IF Y > 191 THEN Y = 191
1490   HPLOT  TO X,Y
1500   IF  ABS ((TGT + 10) - X) < 10 THEN SHOT = SHOT + 1:BANG = 1:HIT = 1
   : REM   HIT
1510   IF FIRE = 0 THEN  GOTO 410
1520   IF FIRE = 5 THEN  GOTO 660
```

衝刺(Sprint)



傾力奔跑的人形圖案

〃百米衝刺〃是一個兩人玩的賽跑遊戲，每一位遊戲者都有一個高解析度人形圖案，當它獲得控制時可橫越螢光幕；控制是藉調整遊戲者的控制器達到的。電腦有一個會隨機來回移動的假想指標，每一位遊戲者都要儘量試圖旋轉各人的控制器以搶先抵達那假想指標；這一段副程式已在第一章中說明過。那一位遊戲者最接近該指標，他就會看到他的跑者衝過螢光幕。

讀取控制器讀數的主迴圈（main loop）在810至890那幾列中。510開始的那幾列內的繪圖副程式是根據一位運動員的一些照片描繪下來；然後轉換爲供HPLOT指令繪圖的X，Y座標。變數MOVEMENT負責掌握跑者的六種姿勢，並在440那一列中輪流呼叫描繪那六種姿勢的副程式：

```
440 ON MOVEMENT GOSUB 510, 570, 610, 660,
710, 760
```

跑者的圖形是本遊戲的一項特色。不過，當您開始要打入本程式並爲程式除錯時，最好先將這些圖形刪去，先檢查程式的其他部份是否可正確執行。只要刪去510至800那幾列，並將440那一列改爲：

```
440 HPLOT X, Y - 20 TO X, Y + 20
```

現在，您看到的不是一個正在跑的人，而是一條移動的垂直線，但這條線會移動得比人形跑者快得多，因爲您省下了高解析度繪圖時間。

Sprint

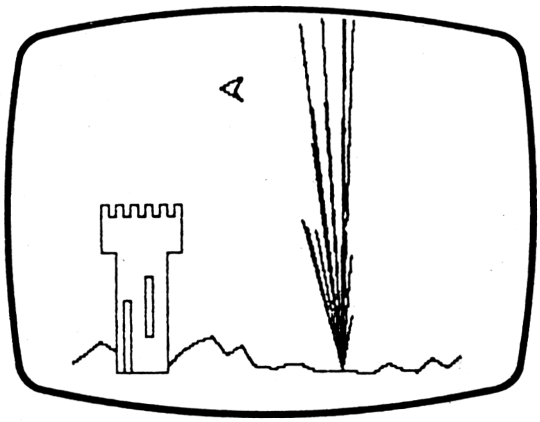
```
10   REM ************************
20   REM **                    **
30   REM **      SPRINT        **
40   REM **                    **
50   REM **        BY          **
60   REM **                    **
70   REM **  STUART ANSTIS     **
80   REM **                    **
90   REM ************************
100   REM
110   REM
120   TEXT : HOME : VTAB 10
130   PRINT "WANT INSTRUCTIONS? (Y/N) "
140   GET Y$
150   IF Y$ = "Y" THEN  GOSUB 1200: REM  INSTRUCTIONS
```

待續...

```
160  HOME : VTAB 10: HTAB 1
170  PRINT "HOW MANY LAPS ? (1--9)"
180  GET NMAX
190  HOME : VTAB 21: HTAB 1
200  PRINT "MAKE SURE THAT BLUE (UPPER) PLAYER HAS ": PRINT "PADDLE # ZER
     O, & RED (LOWER) PLAYER HAS PADDLE # ONE !!"
210  DIM X(2),N(2)
220  DIM H(2),PACE(2)
230 X(1) = 30:X(2) = 30
240 Y(1) = 40:Y(2) = 120
250 N(1) = 0:N(2) = 0
260 R = 80 + 80 *  RND (1)
270  REM : DRAW BORDER ----------
280  HGR
290  HCOLOR= 2
300  HPLOT 2,77 TO 276,77 TO 276,2 TO 2,2 TO 2,77
310  HPLOT 0,79 TO 278,79 TO 278,0 TO 0,0 TO 0,79
320  HCOLOR= 5
330  HPLOT 1,81 TO 1,159 TO 279,159 TO 279,81 TO 1,81
340  HPLOT 3,83 TO 3,157 TO 277,157 TO 277,83 TO 3,83
350  REM : MAIN LOOP ----------
360 H = 3:V =  - 3
370 H(1) = 3:H(2) = 3
380 X = 60
390  FOR MOVEMENT = 1 TO 6
400 PACE(P) = 2 * H(P)
410  GOSUB 810
420  FOR J = 7 TO 0 STEP  - 7
430  HCOLOR= J
440  ON MOVEMENT GOSUB 510,570,610,660,710,760
450  NEXT J
460 X(P) = X(P) + PACE(P)
470  IF X(P) > 250 OR X(P) < 30 THEN N(P) = N(P) + 1:H(P) =  - H(P): GOSUB
     960: REM  PRINT LAP #
480  NEXT MOVEMENT
490  GOTO 390
500  REM  SUBROUTINES 1100 THRU 1600 DRAW THE SIX POSITIONS OF THE RUNNER
     -----
510 X = X + M
520  HPLOT X - 2.5 * H,Y + 2 * V TO X - 2.5 * H,Y + 6 * V TO X + 1.5 * H,
     Y + 6 * V TO X + 2.5 * H,Y + V TO X + 4.5 * H,Y + 4 * V
530  HPLOT X - 6 * H,Y - 9 * V TO X - 7 * H,Y - 7 * V TO X - H,Y - 5 * V TO
     X,Y TO X + 5.5 * H,Y - 2 * V TO X + 4 * H,Y - 8 * V TO X + 6 * H,Y -
     8 * V
540  HPLOT X,Y TO X + 1.5 * H,Y + 6 * V TO X + 1.5 * H,Y + 10 * V TO X +
     3 * H,Y + 10 * V TO X + 4 * H,Y + 7 * V TO X + 1.5 * H,Y + 6 * V
550  RETURN
570  HPLOT X,Y + 3 * V TO X - 2 * H,Y + 5 * V TO X + H,Y + 6 * V TO X + H
     ,Y + 3 * V TO X + 4.5 * H,Y + 4 * V
580  HPLOT X - 7 * H,Y - 4.5 * V TO X - 6 * H,Y - 3 * V TO X - 0.5 * H,Y -
     4 * V TO X,Y TO X + 5 * H,Y - 3.5 * V TO X + 4 * H,Y - 9 * V TO X +
     7 * H,Y - 8 * V
590  HPLOT X,Y TO X + 1.5 * H,Y + 6 * V TO X + 1.5 * H,Y + 10 * V TO X +
     3 * H,Y + 10 * V TO X + 4 * H,Y + 7 * V TO X + 1.5 * H,Y + 6 * V
600  RETURN
610 X = X + M
620  HPLOT X + 1.5 * H,Y + 0.5 * V TO X - 1.5 * H,Y + 3.5 * V TO X + H,Y +
     6 * V TO X,Y + 3 * V TO X + 3 * H,Y + V
630  HPLOT X - 5 * H,Y - 5 * V TO X - 4 * H,Y - 2 * V TO X + H,Y - 5 * V TO
     X,Y TO X + 4 * H,Y - 3.5 * V TO X + 2 * H,Y - 9 * V TO X + 5 * H,Y -
     10 * V
640  HPLOT X,Y TO X + 1.5 * H,Y + 6 * V TO X + 1.5 * H,Y + 10 * V TO X +
     3 * H,Y + 10 * V TO X + 4 * H,Y + 7 * V TO X + 1.5 * H,Y + 6 * V
650  RETURN
660 X = X + 2 * M
670  HPLOT X - H,Y TO X - H,Y + 3.5 * V TO X + 1.5 * H,Y + 6 * V TO X,Y +
     2.5 * V TO X + 3 * H,Y
680  HPLOT X - 3 * H,Y - 5.5 * V TO X - 3 * H,Y - 3 * V TO X + 3 * H,Y -
     4 * V TO X,Y TO X + 2 * H,Y - 4.5 * V TO X - H,Y - 10 * y TO X + 2 *
     H,Y - 10 * V
690  HPLOT X,Y TO X + 1.5 * H,Y + 6 * V TO X + 1.5 * H,Y + 10 * V TO X +
     3 * H,Y + 10 * V TO X + 4 * H,Y + 7 * V TO X + 1.5 * H,Y + 6 * V
700  RETURN
710 X = X + 3.5 * M
720  HPLOT X - 2 * H,Y TO X - 2.5 * H,Y + 4 * V TO X + 1.5 * H,Y + 6 * V TO
     X + 0.5 * H,Y + 2 * V TO X + 4 * H,Y + 2 * V
730  HPLOT X - 2 * H,Y - 10 * V TO X - 5 * H,Y - 10 * V TO X - H,Y - 5 *
     V TO X,Y TO X + 4 * H,Y - 3.5 * V TO X - H,Y - 4.5 * V TO X + 0.5 *
     H,Y - 7 * V
```

```
740  HPLOT X,Y TO X + 1.5 * H,Y + 6 * V TO X + 1.5 * H,Y + 10 * V TO X +
     3 * H,Y + 10 * V TO X + 4 * H,Y + 7 * V TO X + 1.5 * H,Y + 6 * V
750  RETURN
760 X = X + 4 * M
770  HPLOT X - 3 * H,Y + 1.5 * V TO X - 3 * H,Y + 6 * V TO X + 1.5 * H,Y +
     6 * V TO X + 3 * H,Y + 2 * V TO X + 5.5 * H,Y + 4.5 * V
780  HPLOT X - 6.5 * H,Y - 10 * V TO X - 6.5 * H,Y - 7.5 * V TO X - 2 * H
     ,Y - 5 * V TO X,Y TO X + 5 * H,Y - 3 * V TO X + 2.5 * H,Y - 8 * V TO
     X + 5 * H,Y - 9 * V
790  HPLOT X,Y TO X + 1.5 * H,Y + 6 * V TO X + 1.5 * H,Y + 10 * V TO X +
     3 * H,Y + 10 * V TO X + 4 * H,Y + 7 * V TO X + 1.5 * H,Y + 6 * V
800  RETURN
810  REM : PADDLES CHOOSE WHOSE TURN ----------
820 A =  ABS (R -  PDL (0))
830 B =  ABS (R -  PDL (1))
840 P = 1 + (A > B)
850  GOSUB 910
860 R = R + 10 - 20 * ( RND (3) > 0.5): REM  R ALTERS RANDOMLY
870  IF R > 250 THEN R = 220
880  IF R < 10 THEN R = 40
890  RETURN
900  REM :PLAYER P (=1 OR 2) GETS A TURN-----
910 X = X(P)
920 Y = Y(P)
930 PACE = PACE(P)
940 H = H(P)
950  RETURN
960  REM  PRINT LAP #  ----------
970  IF N(1) + N(2) = 1 THEN  HOME
980  VTAB 21 + P: HTAB 1
990  IF P = 1 THEN  PRINT "BLUE (UPPER) ";
1000  IF P = 2 THEN  PRINT "RED (LOWER) ";
1010  PRINT "HAS RUN ";N(P);" LAPS OUT OF ";NMAX;
1020  CALL  - 1059
1030  IF N(P) = NMAX THEN  GOSUB 1050: REM         WINNER
1040  RETURN
1050  REM  WINNER ----------
1060  CALL  - 1059
1070  PRINT
1080  IF P = 1 THEN  PRINT " BLUE (UPPER) ";
1090  IF P = 2 THEN  PRINT " RED (LOWER) ";
1100  PRINT "WINS THE RACE !!!"
1110  PRINT "WANT ANOTHER RACE (Y/N) ?"
1120  GET Y$
1130  IF Y$ = "Y" THEN  HOME : GOTO 230
1140  TEXT : HOME : VTAB 10
1150  PRINT "HIT 'ESC' TO EXIT"
1160  PRINT : PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
1170  GET M$
1180  IF  ASC (M$) = 27 THEN  TEXT : HOME : VTAB 5: PRINT "BYE. THANKS FO
     R PLAYING. ": END
1190  PRINT : PRINT  CHR$ (4);"RUN HELLO"
1200  REM  INSTRUCTIONS ----------
1210  HOME : VTAB 2: GOSUB 1390
1220  PRINT : HTAB 18: PRINT "SPRINT"
1230  PRINT
1240  PRINT : HTAB 1: GOSUB 1390
1250  PRINT : PRINT
1260  PRINT "THIS GAME IS A RACE BETWEEN TWO": PRINT "SPRINTERS.  ";
1270  PRINT "BLUE (UPPER) PLAYER TAKES": PRINT "PADDLE # ZERO, AND RED (L
     OWER) PLAYER": PRINT "TAKES PADDLE # ONE."
1280  PRINT
1290  PRINT "THE COMPUTER HAS AN INVISIBLE POINTER": PRINT "WHICH DRIFTS
     SLOWLY AND RANDOMLY. THE"
1300  PRINT "PLAYER WHOSE PADDLE MATCHES THIS": PRINT "UNSEEN POINTER GET
     S CONTROL, AND ONLY": PRINT "HIS SPRINTER MOVES."
1310  PRINT
1320  PRINT " ********  GOOD LUCK  !!  *******"
1330  PRINT : PRINT
1340  INVERSE
1350  PRINT "        HIT SPACE BAR TO CONTINUE      "
1360  NORMAL
1370  GET K$
1380  RETURN
1390  REM  ASTERISKS -----
1400  FOR J = 1 TO 40: PRINT "*";: NEXT J: RETURN
```

轟炸機(Bomber)



高解析度堡壘，轟炸機和炸彈爆炸圖形

這是一個高解析度圖形的動作遊戲，在遊戲裡，一架轟炸機飛越一片不規則的地形，而在該土地上有一棟孤立的堡壘，您可以隨便利用任何一個鍵來投彈試圖炸毀該堡壘。彈道是一條由地心引力及投彈時飛機速度所決定的拋物線，一旦炸彈離開飛機後，飛機就可以加速或減速，不會影響彈道。當飛機或炸彈碰到螢光幕的左邊線時它會捲回右邊，以稍快的速度和不同的高度從螢光幕右邊重新出現。當您擊中堡壘時它會爆炸，然後又會有一棟新堡壘出現在別的位置，供您做下一波的攻擊。

描繪飛機和炸彈的圖形副程式被放在程式的最上面，甚至在標題之前的 4 至 7 那幾列中，這是爲了確保獲得最快的繪圖速度。100 至 250 那幾列設定變數的數值，260 至 430 那幾列組成使飛機飛行，檢查是否有鍵被按下，以及描繪投下的炸彈的主迴圈。440 至 560 那幾列檢查炸彈落在那裏，如果炸彈擊中堡壘它便產生聲音和閃光，然後擦掉堡壘，再在別的地方重畫一棟新的堡壘供您做下一波攻擊。

Bomber

```
2  GOTO 110
4  REM  PLOT PLANE & BOMB-----
5  HPLOT X,Y TO X + 5,Y - 5 TO X - 10,Y TO X + 5,Y + 5 TO X,Y
6  RETURN
7  HPLOT F,G TO F,G - 5 TO F - 5,G + 5 TO F + 5,G TO F,G: RETURN
10  REM ******************
20  REM *                *
30  REM *     BOMBER     *
40  REM *                *
50  REM *       BY       *
60  REM *                *
70  REM *  STUART ANSTIS *
80  REM *                *
90  REM ******************
100  REM
110  REM
120  TEXT : HOME
130  HGR2 : HGR
140  POKE 230,64: HCOLOR= 5: HPLOT 0,0: CALL 62454: REM  HIDDEN RED EXPLO
     SION FIELD
150  DIM R(40),S(40)
155  GOSUB 2000: REM   RANDOM ARRAY FOR EXPLOSIONS
160  VTAB 21: PRINT "HIT SPACE BAR TO BOMB CASTLE (20 BOMBS)"
180 N = 0: REM  'CLOCK'
190 B = 1:HIT = 1
200 DROP = 1
210 X = 220:Y = 40: REM  PLANE POSITION
220 F = X:G = Y + 10: REM  BOMB POSITION
230 BS = 3: REM  PLANE'S INITIAL SPEED
240  HGR
250 W = 36:WW = 158: GOSUB 800: REM DRAW CASTLE
260  REM  MAIN LOOP -----
270  FOR HC = 7 TO 0 STEP  - 7
280  HCOLOR= HC
290  GOSUB 5: REM DRAW PLANE
300  IF DROP = 2 THEN  GOSUB 7: REM DRAW BOMB
310  NEXT HC
320 X = X - BS: REM  PLANE SPEED INCREASES WITH EACH BOMB DROPPED
330  IF DROP = 2 THEN G = G + N * N / 5
```

```
340  REM  WRAPAROUND:RESET PLANE HEIGHT
350  IF X < 10 THEN X = 275 - RMAX - 22 * H:Y = 40 + 70 *  RND (1)
360  IF DROP = 1 THEN G = Y + 10
370  IF G > 180 THEN  GOTO 440: REM BOMB LANDED ---
380 N = N + 1
390 PRESS =  PEEK ( - 16384)
400  IF PRESS > 127 THEN DROP = 2: POKE  - 16368,0
410 F = X
420  GOTO 270
430  RETURN : REM  DUMMY BOMB SUBROUTINE
440  REM  BOMB LANDS -----
450  VTAB 21: HTAB 1: FOR M = 1 TO 40: PRINT " ";: NEXT M
460  VTAB 21: HTAB 1
470  PRINT "BOMB # ";B;" LANDED -- ";
480  GOSUB 900: REM  EXPLOSION
490 Q = W + 20
500  IF  ABS (F - Q) < 30 THEN  GOTO 570: REM HIT
510  PRINT " YOU MISSED !!"
520 G = 50:N = 0
530 DROP = 1
540 B = B + 1
545 BS = 5 * B: IF BS > 30 THEN BS = 30
550  IF B > 20 THEN  GOTO 1120: REM GAME OVER
560  GOTO 270
570  REM  HIT TARGET -----
580  PRINT " HIT # ";HIT;" !!!"
590  FOR K = 1 TO 5
600  CALL  - 1059
610  POKE  - 16299,0
620  FOR T = 1 TO 20: NEXT T
630  POKE  - 16300,0
640  FOR T = 1 TO 20: NEXT T
650  NEXT K
660 W = 20 + 210 *  RND (1): HGR : GOSUB 800: REM   MOVE HOUSE
670 HIT = HIT + 1
680  GOTO 520
690  END
800  REM  DRAW CASTLE-----
805  GOSUB 1340: REM   DRAW TREES
810  HCOLOR= 7
820  HPLOT W,WW - 50 TO W,WW TO W + 35,WW TO W + 35,WW - 50
830  HPLOT W,WW - 50 TO W - 10,WW - 50 TO W - 10,WW - 70 TO W - 5,WW - 70
      TO W - 5,WW - 65 TO W,WW - 65 TO W,WW - 65
840  HPLOT  TO W,WW - 70 TO W + 5,WW - 70 TO W + 5,WW - 65 TO W + 10,WW -
     65 TO W + 10,WW - 70 TO W + 15,WW - 70 TO W + 15,WW - 65 TO W + 20,W
     W - 65 TO W + 20,WW - 70 TO W + 25,WW - 70
850  HPLOT  TO W + 25,WW - 65 TO W + 30,WW - 65 TO W + 30,WW - 70 TO W +
     35,WW - 70 TO W + 35,WW - 65 TO W + 40,WW - 65 TO W + 40,WW - 70 TO
     W + 45,WW - 70
860  HPLOT  TO W + 45,WW - 50 TO W + 35,WW - 50
870  HPLOT W + 20,WW - 15 TO W + 25,WW - 15 TO W + 25,WW - 40 TO W + 20,W
     W - 40 TO W + 20,WW - 15
880  HPLOT W + 5,WW TO W + 5,WW - 30 TO W + 10,WW - 30 TO W + 10,WW
890  RETURN
900  REM  EXPLOSION-----
910  HPLOT F,WW
960  FOR J = 15 TO 30
970  GOSUB 1060: REM  SOUNDS
980 HC =  INT (7 *  RND (3) + 1)
990  HCOLOR= HC
1000  IF R(J) > F THEN R(J) = F
1010  IF S(J) > WW THEN S(J) = WW
1020  HPLOT F + 10,WW TO F - R(J),WW - S(J)
1030  NEXT J
1040  IF  RND (1) > 0.8 THEN  HGR : GOSUB 800: REM   CLEAR SCREEN
1050  RETURN
1060  REM  SOUNDS-----
1070  FOR M = 1 TO 2
1080 SS =  - 16336
1090 SOUND =  PEEK (SS) +  PEEK (SS) -  PEEK (SS)
1100  NEXT M
1110  RETURN
1120  REM  GAME OVER-----
1130  VTAB 21: HTAB 1
1140  PRINT "YOUR SCORE OF ";HIT;" HITS OUT OF ";B - 1;" IS ";
1150  GOSUB 1250
1160  PRINT "WANT ANOTHER GAME (Y/N) ?"
1170  GET Y$
1180  IF Y$ = "Y" THEN  HOME : GOTO 160
```

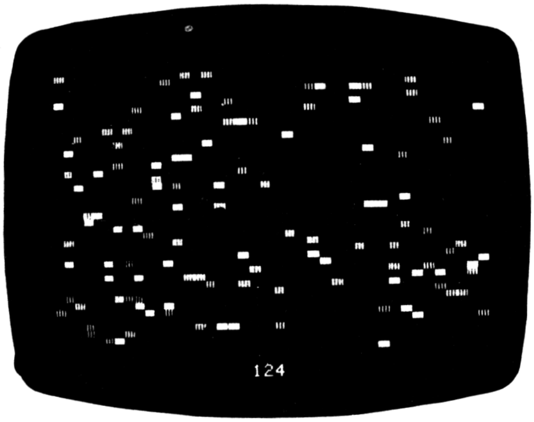
待續……

```
1190  PRINT "TO EXIT, HIT 'ESC'"
1200  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU
1210  GET H$
1220  IF H$ =  CHR$ (27) THEN  TEXT : HOME : VTAB 10: PRINT "OK. THANKS F
     OR PLAYING.": END
1230  PRINT : PRINT  CHR$ (4);"RUN HELLO"
1240  END
1250  REM  COMMENT ON SCORE-----
1260  IF HIT < 2 THEN  PRINT "TERRIBLE! I'M GLAD I'M NOT ON YOUR SIDE.": RETURN
1270  IF HIT < 4 THEN  PRINT "POOR... YOU DIDN'T EVEN FRIGHTEN 'EM!": RETURN

1280  IF HIT < 7 THEN  PRINT "FAIR...TRY AGAIN!": RETURN
1290  IF HIT < 9 THEN  PRINT "QUITE GOOD... TRY AGAIN!": RETURN
1300  IF HIT < 12 THEN  PRINT "GOOD... TRY AGAIN!": RETURN
1310  IF HIT < 15 THEN  PRINT "VERY GOOD...TRY AGAIN!": RETURN
1320  IF HIT < 18 THEN  PRINT : PRINT "EXCELLENT... SHOW 'EM AGAIN!": RETURN

1330  IF HIT < 21 THEN  PRINT : PRINT "OUTSTANDING... YOU SHOULD JOIN THE
      AIR FORCE!!": RETURN
1340  REM PLOTGROUND-----
1350  HGR : HCOLOR= 3
1355 YY = 150
1358  HPLOT W,YY
1360  FOR XX = W TO 0 STEP  - 10
1370  GOSUB 1500
1400  NEXT XX
1405  HPLOT W + 35,YY
1410  FOR XX = W + 35 TO 279 STEP 10
1420  GOSUB 1500
1430  NEXT XX
1450  RETURN
1500 YY = YY + 10 *  RND (3) - 10 *  RND (3)
1510  IF YY > 159 THEN YY = 159
1520  HPLOT  TO XX,YY
1530  RETURN
2000  REM  RANDOM ARRAY FOR EXPLOSION-----
2010  FOR J = 15 TO 30
2020 R(J) = 2 * J *  RND (J) - J
2030 S(J) = 10 * J *  RND (J)
2035  IF RMAX < R(J) THEN RMAX = R(J)
2040  NEXT J
2050  RETURN
```

伺機而發(Wait for It)



使用全部Apple 顏色的五彩碎紙圖案

〞伺機而發〝是一個簡單的〞戰爭心理〝遊戲，每一位遊戲者持着一只遊戲控制器，當大家準備妥當時搶先按下控制器按鈕的人得分——但是，誰等得愈久，累積的時間愈長，誰的分數就愈高，因爲分數是隨着時間累積的。最好的策略是儘可能地等，然後搶在您的對手之前按下按鈕；也就是所謂的〝人不動我不動，人一動我先動〞。

低解析度五彩碎紙圖案與本遊戲的結構沒有什麼關係，它們只是讓您在等待要打敗對手時有些東西可看而已。如果運氣好，它們或許還會令您的對手分心，讓您爭取到寶貴的時間，僥倖地按下按鈕贏得一局。

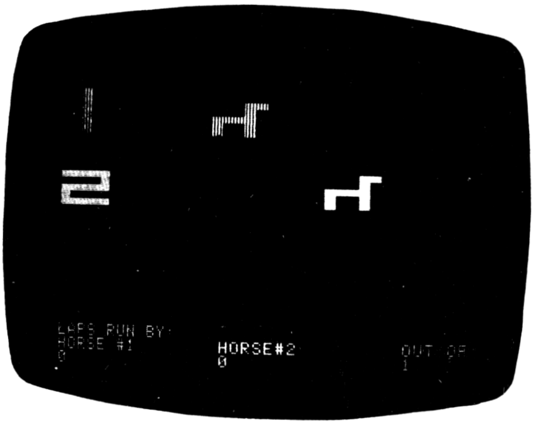
10至250那幾列設定變數數値。260至310那幾列等待遊戲者按下按鈕，附帶用位於610那一列的副程式去描繪隨機低解析度的五彩碎紙圖案，並顯示出大家等著要贏，由一個快速時鐘嘀嗒嘀嗒累積的點數。460至600那幾列宣佈誰是贏家，並邀請遊戲者繼續玩。

Wait For It

```
10   REM *******************
20   REM *                 *
30   REM *    WAIT FOR IT  *
40   REM *                 *
50   REM *        BY       *
60   REM *                 *
70   REM *  STUART ANSTIS  *
80   REM *******************
90   REM
100  REM
105  DIM SUCCESS(2)
110  TEXT : HOME : VTAB 10
120  PRINT "WANT INSTRUCTIONS (Y/N) ?"
130  GET Y$
140  IF Y$ = "Y" THEN  GOSUB 660: REM   INSTRUCTIONS
150  PRINT : PRINT "HOW MANY ROUNDS DO YOU WANT TO PLAY (1--10) ?"
160  INPUT TMAX
170  PRINT : PRINT
180  INVERSE : PRINT "  HIT ANY KEY TO BEGIN  ": NORMAL
190  GET H$
200  REM   INTIALIZE VARIABLES -----
220 SUCCESS(1) = 0:SUCCESS(2) = 0: REM  PLAYERS' SCORES
230 N = 0
240 TURNS = 0
250  GOTO 350
260  REM  MAIN LOOP: WAIT FOR PLAYERS TO PUSH BUTTONS -----
270 N = N + 1
280  FOR P = 0 TO 1
290 X(P) =  PEEK ( - 16287 + P)
300  IF X(P) > 127 THEN SUCCESS(P) = SUCCESS(P) + N: GOTO 460
310  NEXT P
320  GOSUB 610: REM  PLOT
330  GOTO 270
340  REM START NEXT RUN-----
350  TEXT : HOME
360  VTAB 10: PRINT "READY FOR NEXT RUN ?"
370  FOR PAUSE = 1 TO 300: NEXT PAUSE
380  PRINT "STARTING........"
390  FOR PAUSE = 1 TO 500 + 1000 *  RND (1): NEXT PAUSE
400  PRINT : PRINT : PRINT
410  PRINT "  *****      NOW !!!    *****"
420  CALL  - 1059
```

```
430  FOR PAUSE = 1 TO 500: NEXT PAUSE
440  GR
450  GOTO 270
460  REM  ANNOUNCE WHO WON LAST ROUND -----
470  TEXT : HOME
480  VTAB 5: PRINT "ON TURN ";TURNS + 1;" OUT OF ";TMAX;":--"
490  VTAB 10: PRINT "PLAYER ";P;" WON THAT ROUND WITH ";N;" POINTS."
500  PRINT : PRINT
510  FOR P = 0 TO 1: PRINT "PLAYER ";P;"'S TOTAL SCORE IS ";SUCCESS(P);"
     POINTS": PRINT : NEXT P
520 TURNS = TURNS + 1
530  IF TURNS = TMAX THEN 860: REM  GAME OVER
540  INVERSE
550  PRINT : PRINT
560  PRINT : PRINT "HIT SPACE BAR TO CONTINUE"
570  NORMAL
580  GET A$
590 N = 0
600  GOTO 350
610  REM   PLOT RANDOM CONFETTI -----
620  COLOR= 15 *  RND (1) + 1
630 X = 40 *  RND (1):Y = 40 *  RND (1)
640  PLOT X,Y
644  VTAB 22: HTAB 20: PRINT N;"    "
650  RETURN
660  REM   INSTRUCTIONS -----
670  HOME : VTAB 2
680  GOSUB 840
690  PRINT : HTAB 15: PRINT "WAIT FOR IT"
700  PRINT
710  GOSUB 840
720  PRINT : PRINT : PRINT "THIS IS A WAITING GAME."
730  PRINT : PRINT "EACH PLAYER HOLDS A PADDLE.  THE FIRST"
740  PRINT "PLAYER TO PUSH HIS BUTTON WINS -- BUT"
750  PRINT
760  PRINT "THE LONGER YOU WAIT, THE MORE POINTS YOU GET."
770  PRINT : PRINT
780  RETURN
790  INVERSE
800  PRINT "  HIT ANY KEY TO BEGIN...   "
810  NORMAL
820  GET H$
830  RETURN
840  REM  ASTERISKS -----
850  FOR J = 1 TO 40: PRINT "*";: NEXT J: RETURN
860  REM  GAME OVER -----
870  CALL  - 1059: CALL  - 1059
880  INVERSE
890  PRINT "     GAME OVER      "
900  NORMAL
910  PRINT : PRINT "WANT ANOTHER GAME (Y/N) "
920  GET Y$
930  IF Y$ = "Y" THEN  TEXT : HOME : GOTO 150
940  HOME : VTAB 15
950  PRINT "OK. THANKS FOR PLAYING "
960  PRINT : PRINT "WANT TO RETURN TO MENU (Y/N) ?"
970  GET M$
980  IF M$ = "Y" THEN  PRINT  CHR$ (4);"RUN HELLO"
```

賽馬(Horse Race)



兩匹看起來頗像方塊而由控制器驅動的馬

“賽馬”是一個供各年級兒童玩的簡單動作遊戲，遊戲者 INPUT（輸入）他們想要賽跑的圈數（或螢光幕寬度），螢光幕就會出現兩匹以低解析度繪圖法描繪的馬；一匹在上，一匹在下。每一位遊戲者要儘快地來回旋轉他的控制器以驅策他的馬向前飛奔，旋轉較快的控制器將獲得控制權（根據第一章中所說明的方法），並將所屬的馬馳過螢光幕。因此，愈快的遊戲者也終將獲得勝利。

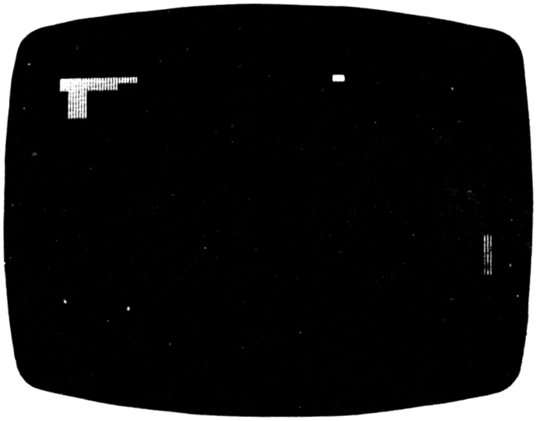
Horse Race

```
10  REM ******************
20  REM *                *
30  REM *   HORSE RACE   *
40  REM *                *
50  REM *       BY       *
60  REM *                *
70  REM *  STUART ANSTIS *
80  REM *                *
90  REM ******************
100 SS(0) = 1:SS(1) = 1
110  FOR P = 0 TO 1:S(P) = SS(P):X(P) = 5:Y(P) = 10 + 10 * P: NEXT P
120  TEXT : HOME : VTAB 10
130  PRINT "WANT INSTRUCTIONS (Y/N)  ?"
140  GET Y$
150  IF Y$ = "Y" THEN  GOSUB 630: REM   INSTRUCTIONS
160  PRINT : INPUT "HOW MANY LAPS (1--8)  ";LMAX
170  HOME : GR
180  GOSUB 780: REM  PLOT DIGITS
190  VTAB 21: PRINT "LAPS RUN BY:";
200  VTAB 22: HTAB 1: PRINT "HORSE #1","HORSE#2","OUT OF:"
210  VTAB 23: PRINT 0,0,LMAX;
220 P = 1
230  REM  MAIN LOOP -----
240 P = 1 - P
250 PAD(P) =  PDL (P)
260  IF  ABS (PAD(P) -  PDL (P)) > S(P) THEN  GOSUB 280: REM  MOVE HORSE
270  GOTO 240
280  REM  MOVE HORSE -----
290  COLOR= 0: GOSUB 420: REM ERASE OLD POSITION
300  COLOR= 9 + P
310 X(P) = X(P) + 1
320  IF X(P) > 35 THEN  GOSUB 360: REM  NEW LAP
330  IF LAP(P) = LMAX THEN 480: REM  RACE WON
340  GOSUB 420: REM  PLOT NEW POSITION
350  RETURN
360  REM  NEW LAP -----
370 X(P) = 6
380 LAPS(P) = LAPS(P) + 1
390  VTAB 23: HTAB 1: PRINT "     ","     ","     ";
400  VTAB 23: HTAB 1: PRINT LAPS(0),LAPS(1),LMAX;
410  RETURN
420  REM  PLOT HORSE -----
430  HLIN X(P),X(P) + 2 AT Y(P)
440  VLIN Y(P) - 2,Y(P) + 2 AT X(P) + 3
450  VLIN Y(P) + 1,Y(P) + 2 AT X(P)
460  PLOT X(P) + 4,Y(P) - 2
470  RETURN
480  REM  RACE WON -----
490  CALL  - 1059: CALL  - 1059
500  HOME
510  VTAB 23: HTAB 1
```

待續……

```
520  PRINT "HORSE # ";P + 1;" IS THE WINNER!!"
530  PRINT "WANT ANOTHER RACE (Y/N)?";
540  GET Y$
550  IF Y$ = "Y" THEN 590
560  TEXT : HOME : VTAB 10
570  PRINT "OK.  THANKS FOR PLAYING."
572  PRINT : PRINT "HIT 'ESC' TO EXIT"
574  PRINT : PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
576  GET K$
578  IF  ASC (K$) = 27 THEN  HOME : END
580  GOTO 900
590 SS(P) = SS(P) + 1:  REM  HARDER FOR WINNER NEXT TIME
600 X(0) = 5:X(1) = 5
610 LAPS(0) = 0:LAPS(1) = 0
620  GOTO 170
630  REM   INSTRUCTIONS -----
640  HOME : VTAB 5
650  PRINT "                HORSE RACE"
660  PRINT : PRINT
670  PRINT "****************************************"
680  PRINT : PRINT
690  PRINT "FOR TWO PLAYERS."
700  PRINT
710  PRINT "TURN YOUR PADDLE RAPIDLY TO AND FRO TO"
720  PRINT "URGE YOUR HORSE ALONG."
730  PRINT
740  PRINT "      HIT ANY KEY TO CONTINUE."
750  PRINT : PRINT
760  GET H$
770  RETURN
780  REM   PLOT DIGITS -----
790  COLOR= 9
800 X = 2:Y = Y(0)
810  VLIN Y - 4,Y + 2 AT X
820  COLOR= 10
830 Y = Y(1) + 2
840  FOR C = 0 TO 4 STEP 2
850  HLIN 0,4 AT Y - C
860  NEXT C
870  VLIN Y - 2,Y AT 0
880  VLIN Y - 4,Y - 2 AT 4
890  RETURN
900  REM  RETURN TO MENU -----
910  PRINT
920  PRINT  CHR$ (4);"RUN HELLO"
930  END
```

打靶(Target)



移動的靶標和發射的子彈

“打靶” 是一個射擊練習遊戲，有一個靶標在螢光幕的右邊不斷地上下移動，您從左邊以一把手槍對準它射擊，您的得分與失分將會在每次射擊之後顯示出來。為了使遊戲更困難，更刺激，靶標移動的速度與手槍的位置在每次射擊之後都會改變。

遊戲者 INPUT 他所希望的靶標大小（TGT）；明顯地，靶標愈小遊戲愈困難。140 至 250 那幾列描繪手槍圖形。110 至 148 設定變數的初值，包括靶標移動速度XSPD；其數值在 250 那一列中設定為介於 1 與 9 之間的亂數。 260 至 380 那幾列構成主迴圈，令靶標上下移動；320 那一列描繪一條垂直靶線，其長度由變數 TGT 所決定；而 340 和 350 那兩列則在其上下畫一條黑線將靶標的前一個位置擦去。

當遊戲者隨便按一個鍵時 370 那一列便會跳到 390 至 470 那幾列的副程式去發射一發子彈；如果子彈擊中靶標（ 520 那一列），螢光幕便會顯示一段 “You

hit it !!″（您擊中了）訊息，並將積分加 1（580 至 630 那幾列）。按下去，手槍跳到新位置，靶標也另換一個速度，遊戲又重頭開始。

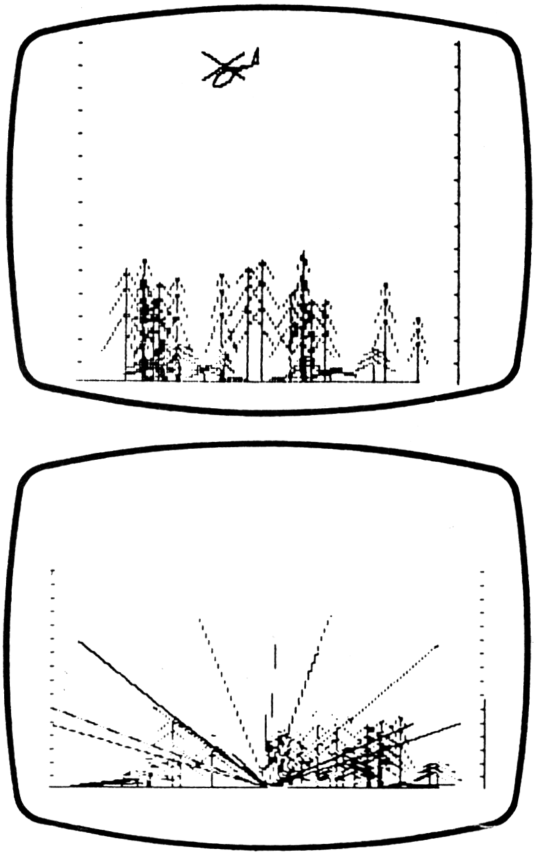
Target

```
10  REM ********************
20  REM *                  *
30  REM *      TARGET      *
40  REM *                  *
50  REM *        BY        *
60  REM *                  *
70  REM *  STUART ANSTIS   *
80  REM *                  *
90  REM ********************
100 W$(0) = "BAD":W$(1) = "POOR":W$(2) = "FAIR":W$(3) = "GOOD":W$(4) = "G
    REAT!!"
110  TEXT : HOME : VTAB 22
120  PRINT "HIT SPACE BAR TO SHOOT TARGET (20 SHOTS)"
130  GR
140  REM  DRAW PISTOL -----
150  COLOR= 12
160 P = 34 *  RND (1) + 1
170  HLIN 0,7 AT P
180  HLIN 0,5 AT P + 1
190  FOR J = 1 TO 2: VLIN P + 1,P + 5 AT J: NEXT J
200  COLOR= 0
210  FOR K = 0 TO 7: VLIN 0,P - 1 AT K: VLIN P + 6,39 AT K: NEXT K
220 X = 20
230 SOLD = 8
240 SSPD = 3
250 XSPD = 3 *  RND (1)
260  REM  DRAW TARGET -----
270 X = X + XSPD
280  IF X > 1 AND X < 35 THEN  GOTO 310
290 XSPD =  - 1 * XSPD
300  GOTO 270
310  COLOR= 1
320  VLIN X,X + 5 AT 39
330  COLOR= 0
340  VLIN 0,X - 1 AT 39
350  VLIN X + 6,39 AT 39
360 KEY =  PEEK ( - 16384)
370  IF KEY > 127 THEN  GOTO 400
380  GOTO 270
```

待續…

```
390  REM  SHOOT -----
400 SN = SOLD + SSPD
410  COLOR= 15
420  PLOT SN,P
430  COLOR= 0
440  PLOT SOLD,P
450 SOLD = SN
460  IF SN > 37 THEN  GOTO 480
470  GOTO 270
480  REM  CHECK FOR HIT -----
490  COLOR= 0: PLOT SN,20
500 SHOTS = SHOTS + 1
510  IF SHOTS > 19 THEN  GOTO 690
520  IF  SCRN( 39,P) <  > 0 THEN  GOSUB 580
530 SN = 0
540  HOME
550  VTAB 21: HTAB 20: PRINT "SCORE: ";HITS;" OUT OF ";SHOTS
560  POKE  - 16368,0
570  GOTO 120: REM  WAS 10 ***********
580  REM  HIT -----
590  VTAB 21
600  PRINT "YOU HIT IT !!!"
610 HITS = HITS + 1
620  GOSUB 650
630  RETURN
640  REM  SOUNDS -----
650  FOR M = 1 TO 40:SOUND =  PEEK ( - 16336): NEXT M
660  FOR TIME = 1 TO 100: NEXT TIME
670  VTAB 22: PRINT "                                    "
680  RETURN
690  REM  END OF GAME -----
700  FOR J = 1 TO 3: GOSUB 640: NEXT J
710  HOME : VTAB 22
720 K =  INT (HITS / 4.5)
730  PRINT "YOUR SCORE OF ";HITS" OUT OF 20 IS ";W$(K)
740  VTAB 23: HTAB 1: PRINT "WANT ANOTHER GAME (Y/N) ?";
750  GET YN$
760  GET YN$
770  IF YN$ = "Y" THEN SHOTS = 0:HITS = 0: GOTO 110
780  VTAB 24: HTAB 1: PRINT "HIT 'ESC' TO EXIT"
790  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU   ";
800  GET H$
810  IF  ASC (H$) = 27 THEN  TEXT : HOME : END
820  PRINT
830  PRINT  CHR$ (4);"RUN HELLO"
840  END
```

直昇機(Helicopter)



直昇機在一片高解析度森林上翺翔

這是一個很古老，但很受人喜愛的遊戲——係有名的Lunar Lander（月球登陸隊）的一種版本，它原先是以Thomas A. Dwyer和Margot Critchfield合著的BASIC and the Personal Computer（BASIC與個人電腦）一書中所寫的Planet-X Lander（X星球登陸隊）爲藍本的。不過，它已被廣泛地改寫爲Apple程式，並被改寫爲以高解析度圖形處理的交互作用式(interactive)遊戲。

這是一個模擬遊戲，在遊戲中您要想辦法控制直昇機使它很優美地降落在地上。您要藉調整您的油門（0號遊戲控制器）來改變上昇的推力，而地心引力會把直昇機拉下來，因此除非您使用油門使直昇機上昇，否則直昇機會下降得愈來愈快，也就是會有一個下降加速度，到最後直昇機終將墜毀。另一方面，如果您使用了太大的推力來反抗地心引力下拉的力量，您可能會向上飛，更可怕的是您可能會在半空中耗光油料，那就是您的末日到了！您的目標是要緩慢地降落，要以低於每秒5公尺（大約每小時10哩）的垂直速度下降。

本遊戲分成八個主要部份。

1. 設定初值(80至180那幾列)。直昇機是由130公尺開始，因此HEIGHT=130；它重5000公斤，因此S = 5000；機上有15,000公斤油料，因此E=15,000；飛行由時間零開始，因此T=0。在本遊戲中，動力加速度僅每秒每秒2.4公尺（大約地球的¼）。
2. 220那一列會在螢光幕底部顯示一份進行狀況報告，隨時讓您知道速度、高度、推力和剩餘的油料。此外，還有一個油表在螢光幕右邊，用一條垂直紅線來指示剩餘的油量。您也可以看著直昇機在螢光幕上飛行，利用它來判斷直昇機的高度和下降速率。
3. 200那一列讀取您的控制器讀數，並用它代表上昇推力的大小（介於0與250之間）。油料在270那一列中以每單位推力消耗1公斤的速率減少。
4. 新的速度和高度是在290和320那兩列中計算的，根據牛頓第二定律我們知道：速度改變量／時間改變量＝（$-G+F/M$）；此處，G＝重力加速度，而F/M＝火箭推力所產生的向上加速度。質量包含兩部分：直昇機S和油料E，所以當您用完油料時質量會減少。

5. 340 那一列檢查是否發生撞擊，而 330 那一列檢查油料是否耗盡，如果其中沒有一項爲眞，程式便再回去讀取油門的讀數。

6. 370 至 410 那幾列處理油料耗盡的情況，它包含呼叫在螢光幕畫出墜毀畫面的副程式。

7. 430 那一列判斷您是否做了漂亮的著陸（應小於－5 公尺／秒），440 那一列檢查差勁但安全的著陸動作（介於－5 與－15 公尺／秒之間），但是如果您的降落速度快過這個數值，則 450 那一列會直接跳到墜毀圖形副程式。

8. 繪圖部份包括一段您每次玩這遊戲時都會描繪一片明顯不同森林的副程式。事實上，每一顆樹都是一樣的，只是程式每畫一顆樹都是隨機決定它的形狀（高度與寬度），使它看起來像一片不同的樹林；自然界差不多也就是這樣子。

當直昇機接近地面時會變得愈來愈大，這是因爲 730 至 900 那幾列的直昇機繪圖副程式包含一個水平比例因素 H 和一個垂直比例因素 V，它們的初值是在 90 那一列內設定的，然後在 840 和 850 那兩列變得愈來愈大。注意，H 和 XSPD 也在 840 和 850 那兩列變爲－H 和－XSPD，這是表示直昇機向右飛過螢光幕，然後又折回飛向左邊。

Helicopter

```
10  REM  *************************
20  REM          HELICOPTER
25  REM
30  REM  *************************
40  REM
50  REM  AFTER CRUTCHFIELD & DWYER ---
60  TEXT : HOME
70  VTAB 22: PRINT "SPEED      HEIGHT      THRUST      FUEL"
80  GOSUB 1030: REM  DRAW TREES -----
90 H = 1:V = 1: REM  SCALE OF HELICOPTER
100  SPIN = 1: REM  POSITION OF HELICOPTER ROTOR
110  XSPD =  - 10: REM  SIDEWAYS MOTION OF HELICOPTER
120  X = 100
130  HEIGHT = 130
140  S = 5000
150  E = 15000: REM  AMOUNT OF FUEL
160  T = 0
170  G = 2.4
180  VEL =  - 100
190  A =  INT (HEIGHT + 0.5)
200  F =  INT (1.9 *  PDL (0))
210   VTAB 21: PRINT "                                                   "
220   VTAB 21: PRINT  INT (VEL); TAB( 15); INT (HEIGHT); TAB( 25);F; TAB(
     36);E
230  FUEL = 159 - E / 100: REM   CONVERT FUEL FOR PLOTTING
240  Y = 159 - HEIGHT
250   IF Y < 20 * V THEN Y = 20 * V
260   GOSUB 690: REM   GRAPHICS SUB ROUTINE
270  E = E - F
280  V1 = VEL
290  V2 = 2000 * F / (S + E) - 10 * G
300  VEL = V1 + V2
310  D = (V1 + V2 / 2) * 10
320  HEIGHT = HEIGHT + D / 1000
330   IF E <  = 0 THEN 370
340   IF HEIGHT < 0 THEN  GOTO 430
350  T = T + 10
360   GOTO 190
370   PRINT "YOU'RE OUT OF FUEL ";A;" KILOMETERS UP    "
380   FOR M = 1 TO 50:BUZZ =  PEEK ( - 16336): NEXT M
390   FOR PAUSE = 1 TO 1000: NEXT PAUSE
400   GOSUB 590: REM   CRASH GRAPHICS
410   GOTO 550
420  F = 0
430   IF VEL >  =  - 5 THEN 500: REM   BEAUTIFUL LANDING
440   IF VEL >  =  - 15 THEN 530: REM  CLUNK
450   GOSUB 590: REM  CRASH GRAPHICS
460   PRINT "CRUNCH!!! VELOCITY ="; INT (VEL * 100) / 100;" M/SEC"
470   PRINT "THAT'S "; INT (100 * VEL * 3.2 * 3600 / 5280) / 100;" MPH ==
     OH BOY !"
480   PRINT "TURN IN YOUR LICENCE AT ONCE "
490   GOTO 550
500   PRINT "BEAUTIFUL LANDING *** VELOCITY = ";VEL * 3.2 * 3600 / 5280;"
     MPH"
510   PRINT "YOU HAD ";E;" KILOGRAMS OF FUEL LEFT"
520   GOTO 550
530   PRINT "CLUNK --RATTLE-RATTLE --SQUINCH"
540   PRINT "V= ",VEL * 3.2 * 3600 / 5280;" MPH:ROUGH, BUT YOU SURVIVED"
550   PRINT "WANT ANOTHER FLIGHT (Y/N) ",
560   GET Y$
570   IF Y$ = "Y" THEN  GOTO 60
```

```
572  PRINT "WANT TO RETURN TO MENU (Y/N ?"
574  GET M$
576  IF M$ = "Y" THEN  TEXT : PRINT : PRINT  CHR$ (4);"RUN HELLO"
580  END
590  REM  CRASH GRAPHICS -----
600  HOME : VTAB 21
610  FOR I = 1 TO 30
620  HCOLOR= 8 *  RND (1)
630 C = 279 *  RND (1)
640 D = 159 *  RND (1)
650  HPLOT 140,158 TO C,D
660  FOR M = 1 TO 5:SOUND =  PEEK ( - 16336): NEXT M
670  NEXT I
680  RETURN
690  REM : HELICOPTER 1 -----
700  REM  PLOT FUEL GAUGE -----
710  HCOLOR= 0: HPLOT 278,0 TO 278,FUEL
720  HCOLOR= 7: HPLOT 278,FUEL TO 278,159
730  REM  PLOT HELICOPTER -----
740  FOR J = 7 TO 0 STEP  - 7
750  FOR M = 1 TO 5:BUZZ =  PEEK ( - 16336): NEXT
760  HCOLOR= J
770  HPLOT X + 7 * H,Y - 9 * V TO X + 7 * H,Y - 7 * V TO X + 12 * H,Y - 8
      * V TO X + 28 * H,Y - 10 * V TO X + 32 * H,Y - 17 * V TO X + 32 * H
     ,Y - 10 * V TO X + 15 * H,Y - 6 * V TO X + 12 * H,Y - 2 * V TO X + 6
      * H,Y
780  HPLOT  TO X,Y TO X + H,Y - 3 * V TO X + 3 * H,Y - 5 * V TO X + 8 * H
     ,Y - 7 * V TO X + 8 * H,Y - 9 * V
790  ON SPIN GOSUB 910,950,990
800  NEXT J
810 ROOM = 35 *  ABS (H)
820  IF X > ROOM AND X < 279 - ROOM THEN  GOTO 870
830 XSPD =  - 1.1 * XSPD
840 H =  - 1.05 * H
850 V = 1.05 * V
860  REM  Y=40*V
870 X = X + XSPD
880 SPIN = SPIN + 1
890  IF SPIN > 3 THEN SPIN = 1
900  RETURN
910  REM    SUBROUTINE FOR SPIN=1 -----
920  HPLOT X - 8 * H,Y - 15 * V TO X + 22 * H,Y - 3 * V
930  HPLOT X - 8 * H,Y - 3 * V TO X + 22 * H,Y - 15 * V
940  RETURN
950  REM    SUBROUTINE FOR SPIN=2 -----
960  HPLOT X - 10 * H,Y - 7 * V TO X + 24 * H,Y - 11 * V
970  HPLOT X,Y - 19 * V TO X + 15 * H,Y + 2 * V
980  RETURN
990  REM    SUBROUTINE FOR SPIN=3 -----
1000  HPLOT X - 10 * H,Y - 11 * V TO X + 24 * H,Y - 7 * V
1010  HPLOT X,Y - V TO X + 14 * H,Y - 19 * V
1020  RETURN
1030  HGR
1040  REM  DRAW TREES -----
1050  GOSUB 1140
1060 YY = 158
1070  FOR I = 1 TO 30
1080 HH = 1 + 2 *  RND (1)
1090 VV = 3 *  RND (1)
1100 XX = 25 + 225 *  RND (2)
1110  GOSUB 1180
1120  NEXT I
1130  RETURN
1140  REM    PLOT GROUND -----
1150  HCOLOR= 1: HPLOT 0,158 TO 250,158
1160  FOR I = 0 TO 159 STEP 10: HCOLOR= 7: HPLOT 275,I TO 277,I: NEXT I
1170  FOR I = 0 TO 159 STEP 10: HCOLOR= 7: HPLOT 2,I TO 4,I: NEXT I
1180  REM    PLOT TREE -----
1190  HCOLOR= 7
1200  HPLOT XX,YY TO XX,YY - 19 * VV
1210  HCOLOR= 1
1220  HPLOT XX - 7 * HH,YY - 5 * VV TO XX,YY - 12 * VV TO XX + 7 * HH,YY -
     5 * VV
1230  HPLOT XX - 5 * HH,YY - 11 * VV TO XX,YY - 16 * VV TO XX + 5 * HH,YY
      - 11 * VV
1240  HPLOT XX - 3 * HH,YY - 16 * VV TO XX,YY - 19 * VV TO XX + 3 * HH,YY
      - 16 * VV
1250  RETURN
```

緊迫盯人(Tiddler)

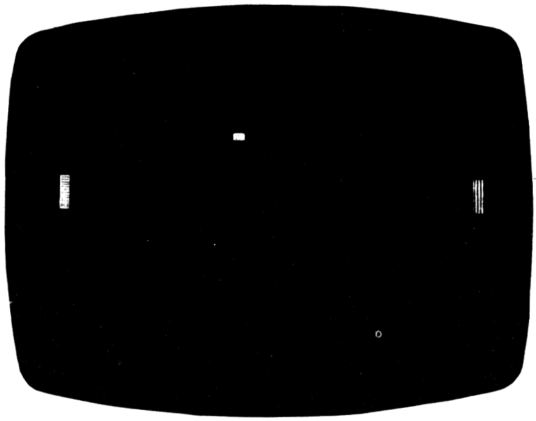
在這個遊戲中，每位遊戲者都要持著一個遊戲控制器不斷地調整，以便獲得控制權。電腦有一個隨機來回移動的假想指標，遊戲者要儘量調整控制器去盯著該指標，誰最靠近該指標誰就獲得控制權。當左方遊戲者獲得控制權時有一條水平彩色線條會出現在螢光幕左邊，線愈長表示當時他愈接近指標，他所獲得的點數也就愈多；右手邊的情況也是一樣的，當時間停止時，每位遊戲者的分數會自動 PRINT 出來。

這是玩“衝刺”遊戲（ Sprint ）之前很好的練習遊戲；它們使用相同的控制器控制方法，我們已在第一章說明過。

Tiddler

```
800  REM  TOM TIDDLER'S GROUND -----
810  DIM SUCCESS(2)
820  PRINT : PRINT : PRINT
850 SUCCESS(1) = 0:SUCCESS(2) = 0: REM  PLAYERS' SCORES
860 N = 0:Y = 0
870  GR
900 R = 255 *  RND (1)
2000  REM : PADDLES CHOOSE WHOSE TURN ----------
2002 N = N + 1
2004  IF N > 200 THEN  GOTO 3000: REM    END OF GAME
2010 A =  ABS (R -  PDL (0))
2020 B =  ABS (R -  PDL (1))
2030  IF A < B THEN P = 1: GOSUB 2200
2040  IF A > B THEN P = 2: GOSUB 2200
2042  IF A = B THEN R = R + 10:C =  RND (1): IF C > 0.5 THEN R = R - 10: GOTO
     2010
2050 C =  RND (1)
2060  IF C > 0.5 THEN R = R - 10: GOTO 2080
2070 R = R + 10
2080  IF R > 250 THEN R = 220
2090  IF R < 10 THEN R = 40
2110  GOTO 2000
2190  REM :PLAYER P (=1 OR 2) GETS A TURN-----
2200 E = (A - B) / 5
2210  IF E > 18 THEN E = 18
2215  IF E <  - 18 THEN E =  - 18
2220 SUCCESS(P) = SUCCESS(P) +  ABS (E)
2230 Y = Y + 1
2240  IF Y > 39 THEN Y = 0: GR
2245 CC = 15 *  RND (1) + 1
2250  COLOR= CC
2260  HLIN 19 + P,19 + P + E AT Y
2270  COLOR= 9
2280  PLOT 20,Y
2290  RETURN
3000  REM  WINNER ----------
3001  PRINT "GAME OVER"
3002  FOR PAUSE = 1 TO 1500: NEXT PAUSE
3004  TEXT : HOME : VTAB 10
3005  FOR P = 1 TO 2
3010  PRINT "PLAYER ";P;" SCORED "; INT (SUCCESS(P))
3012  PRINT
3015  NEXT P
3018  PRINT
3020 TRUE = SUCCESS(1) < SUCCESS(2)
3025 WIN =  ABS ( INT (SUCCESS(1) - SUCCESS(2)))
3030  PRINT "PLAYER # ";TRUE + 1;" IS THE WINNER ";
3032  PRINT "BY ";WIN;" POINTS !"
3035  PRINT : PRINT
3040  PRINT "WANT ANOTHER GAME (Y/N) ?"
3050  GET Y$
3070  IF Y$ = "Y" THEN  PRINT : PRINT : PRINT : PRINT : GOTO 860
3080  TEXT : HOME
3090  END
```

乒乓(Ping Pong)



電動遊戲的老祖宗

這是一個用低解析度圖形繪製的電腦乒乓球遊戲，它是由第一章中說明的〝乒乓球拍〞（ Pongbat ）副程式與Applesoft Tutorial 一書內的〝跳動的球〞（ Bouncing Ball ）副程式所組成。

本遊戲可供一人或兩人玩。40那一列詢問您要〝SQUASH(1) OR TENNIS (2)?〞，如果您要兩個人玩，您應該輸入一個2，然後400至490那幾列內的乒乓球拍副程式將會供給左方遊戲者一個黃色拍子，供給右方遊戲者一個藍色拍子，每一個拍子各由一個遊戲控制器控制。一個人的時候應該輸入一個1，410和480那兩列便會在左邊重疊一個紅拍子，留下一個藍拍子在右邊供單人遊玩。

70那一列詢問您要〝COOPERATIVE (1) OR COMPETITIVE (2)〞，如果您輸入2，您所得到的是正常的遊戲，但是，如果您輸入的是1，您將會得到一個特別設計，特別容易玩的〝人道主義〞乒乓球；當球到達時如果您的拍子位置不對，球會一直等到您的拍子到達，然後才自動彈離拍子！這是由230至250那

幾列中的

```
SCRN (XOLD + SGN(XSPEED), YNEW)
```

處理的。

這幾列監視緊接著球前面的小方塊的螢光幕顏色，如果拍子在那兒，小方塊的COLOR 必不爲零，因此球將會在 330 那一列彈回去；如果拍子不在那兒，則COLOR 將爲零，程式便一直等待到拍子到達爲止。

如果 XSPEED 一直爲 ＋1 ，則 230 那一列的運算式便可簡化爲：

```
SCRN (XOLD+XSPEED, YNEW)
```

事實上，SGN(XSPEED)用在此處是因爲如果 XSPEED 爲正，其值爲1；如果 XSPEED＝0，其值爲 0；如果 XSPEED爲負，則其值爲－1，解決了XSPEED不等於＋1 的情況。在本程式中，當球向右飛時 XSPEED等於 ＋3 ，而當它向左飛時 XSPEED 等於 －3 。若想使遊戲更快或更慢，只要增加或減少160 那一列中XSPEED 的數值。

乒乓可說是電視遊戲的老祖宗。上述的“人道主義”乒乓對不熟悉電腦遊戲，很容易緊張的人而言，無疑是艘很有用的“破冰船”，它是那些深恨電視遊樂器的人的電視遊戲。

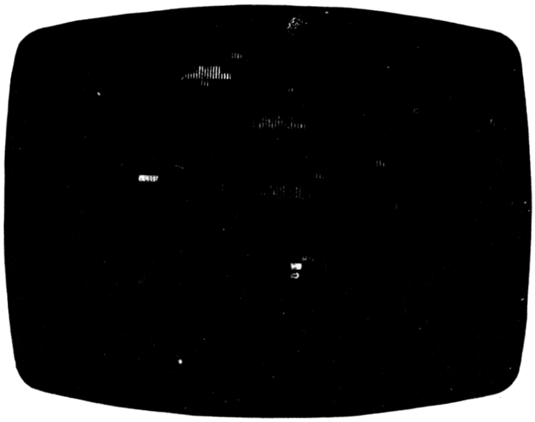
Ping Pong

```
10   REM PING PONG----
20   TEXT : HOME : VTAB 5
30   PRINT "HUMANISTIC PING PONG:": PRINT
40   PRINT "SQUASH (1) OR TENNIS (2)...";
50   GET SQUASH
60   PRINT
70   PRINT "CO-OPERATIVE (1) OR COMPETITIVE (2) ";
80   GET CO
90   PRINT
100   PRINT "BALL SPEED: SLOW (1), MEDIUM (2),"
110   PRINT "FAST (3)  "
120   GET XSPD
130 YSPD = 1
140   PRINT
150   GR
```

待續

```
160  REM BOUNCE BALL-----
170 XOLD = 20
180 YOLD =  INT (35 *  RND (1))
190 XN = XOLD + XSPD
200  GOTO 400
210  IF (XN > 1) AND (XN < 39) THEN  GOTO 280
220  REM CHECK IF PADDLE HITS BALL-----
230 SS =  SCRN( XOLD +  SGN (XSPD),YN)
240  IF SS = 0 AND CO = 2 THEN  GOTO 500
250  IF SS <  > 0 THEN  GOTO 370
260 XN = 20
270  GOTO 190
280 YN = YOLD + YSPD
290  IF (YN > 0) AND (YN < 35) THEN  GOTO 330
300 YSPD =  - 1 * YSPD
310  GOTO 280
320  REM PLOT BALL-----
330  COLOR= 15: PLOT XN,YN
340  COLOR= 0: PLOT XOLD,YOLD
350 XOLD = XN:YOLD = YN
360  GOTO 190
370 CLICK =  PEEK ( - 16336) =  PEEK ( - 16336)
380 XSPD =  - XSPD:XN = XN + XSPD: GOTO 280
390  REM  MOVE PADDLES-----
400 PAD(P) =  INT ( PDL (P) / 8)
410  IF SQUASH = 1 AND P = 0 THEN 480
420  IF PAD(P) < 1 THEN 440
430  COLOR= 0: VLIN 0,PAD(P) - 1 AT 38 * P + 1
440  COLOR= 0: VLIN PAD(P) + 5,39 AT 38 * P + 1
450  COLOR= 13 - 7 * P: VLIN PAD(P),PAD(P) + 4 AT 38 * P + 1
460 P = 1 - P: REM  CHANGE PLAYER
470  GOTO 210
480  COLOR= 9: VLIN 0,35 AT 1
490  GOTO 460
500  REM MISS BALL---
510  CALL  - 1059
520  COLOR= 0: PLOT XOLD,YOLD
530  HOME : PRINT "HIT ANY KEY FOR NEXT BALL"
540  GET G$
550  GR : HOME : GOTO 160
```

太空突擊隊(Space Zapper)



襲擊入侵者，但小心太空雷

這是一個供一人或兩人玩的低解析度動作遊戲，每位遊戲者利用遊戲控制器操縱他的太空船，並根據打下紅色碟形基地的多寡來累計分數。遊戲者必須衝進基地“吃”掉它，才能得分，然後回頭再去吃其他基地。同時，遊戲者還必須提防那些靜止的藍色太空雷，因爲誤觸一枚太空雷會使分數自動減半。爲了使遊戲更刺激，基地會在太空中不定時變換位置；此外，它們有時還會變成藍色，因此十分危險，因爲它們那時是一個巨大的機動太空雷。碰到變成藍色的基地也會使您的分數減半，如果您那時衝過三、四格，則您的分數會連續很快地減半三、四次；那可是很慘的事。所幸的是您不必擔心是否會撞到另一位遊戲者，您可以逕行飛過另一位遊戲者而不會有什麼損失。

您需要一點時間去習慣太空船的操縱；將您的控制器看成好像有一個指標從2點鐘方向開始，可順時鐘轉到12點鐘方向，您的太空船將會朝控制器指標所指的方向移動。如果把控制器刻畫成0到7，則您移動的方向（由“*”開始）

將為：

```
6 7 0
5 * 1
4 3 2
```

您一定會知道您移動的方向，因為您的太空船包含兩個小方塊：一個白色的在前面指向您要移動的方向，好像船頭一樣；另一個彩色小方塊在後面像船尾一樣。

Space Zapper

```
10   REM ************************
20   REM *                      *
30   REM *      SPACE ZAPPER    *
40   REM *                      *
50   REM *                      *
80   REM ************************
90   REM
100  REM
110  TEXT : HOME
120  VTAB 10
130  PRINT "HOW MANY PLAYERS -- 1 OR 2 ?"
140  GET N
150  IF N <  > 1 AND N <  > 2 THEN 130
160  VTAB 15: PRINT "WANT INSTRUCTIONS (Y/N) ?"
170  GET Y$
180  IF Y$ = "Y" THEN  GOSUB 1020: REM  INSTRUCTIONS
190 PMAX = N - 1
200  VTAB 21: HTAB 10: PRINT "SCORE    TIME";
210  GR
220  GOSUB 720: REM  PLOT KLINGONS
230  GOSUB 840: REM  PLOT SPACE MINES
240 X(0) = 15:Y(0) = 20
250 X(1) = 30:Y(1) = 20
260  REM  MAIN LOOP -----
270 P = 0
280 TIME = TIME + 1
290  IF TIME / 5 =  INT (TIME / 5) THEN  GOSUB 560: REM MOVE KLINGON
```

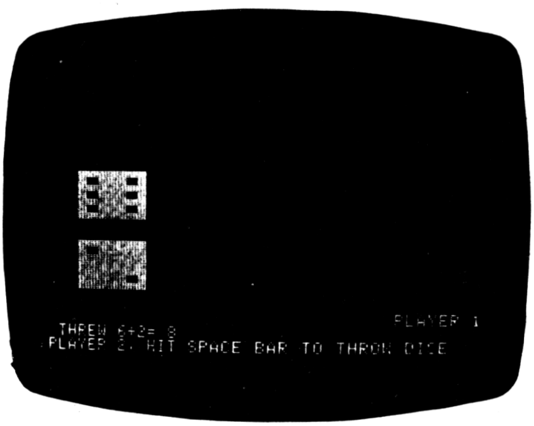
待續

```
300  COLOR= 0
310  PLOT XOLD(P),YOLD(P)
320  PLOT X(P),Y(P): REM  ERASE OLE POSITION
330 XOLD(P) = X(P):YOLD(P) = Y(P)
340  GOSUB 430: REM   STEER
350  IF  SCRN( X(P),Y(P)) = 2 THEN  GOSUB 910: REM  HIT A SPACE MINE
360  IF  SCRN( X(P),Y(P)) = 9 THEN  GOSUB 970: REM  HIT A KLINGON
370  COLOR= 15
380  PLOT X(P),Y(P): REM PLOT NEW POSITION
390  COLOR= 11 + P
400  PLOT XOLD(P),YOLD(P)
410 P = PMAX - P
420  GOTO 280
430  REM  PADDLE STEERS -----
440 J(P) =  INT ( PDL (P) / 31.8)
450  VTAB 22 + P: HTAB 10: PRINT "
460  VTAB 22 + P: HTAB 10: PRINT WIN(P); SPC( 9);: IF P = 0 THEN  PRINT T
     IME
470  IF J(P) < 2.5 THEN X(P) = X(P) + 1
480  IF J(P) > 3.5 AND J(P) < 6.5 THEN X(P) = X(P) - 1
490  IF J(P) > 1.5 AND J(P) < 4.5 THEN Y(P) = Y(P) + 1
500  IF J(P) > 5.5 OR J(P) < 0.5 THEN Y(P) = Y(P) - 1
510  IF X(P) > 39 THEN X(P) = 39
520  IF X(P) < 0 THEN X(P) = 0
530  IF Y(P) > 39 THEN Y(P) = 39
540  IF Y(P) < 0 THEN Y(P) = 0
550  RETURN
560  REM  : MOVE A KLINGON -----
570 Q =  PEEK ( - 16336) +  PEEK ( - 16336)
580 N =  INT (3 *  RND (1) + 0.5)
590 RAN =  RND (1)
600  COLOR= 0
610  GOSUB 790: REM  ERASE OLD KLINGON POSITION
620 R(N) = R(N) + (RAN > 0.5) - 2 * (RAN > 0.75)
630 S(N) = S(N) + (RAN < 0.5) - 2 * (RAN < 0.25)
640  IF R(N) > 35 THEN R(N) = R(N) - 10 *  RND (1)
650  IF R(N) < 0 THEN R(N) = 10 *  RND (1)
660  IF S(N) > 35 THEN S(N) = S(N) - 10 *  RND (1)
670  IF S(N) < 0 THEN S(N) = 10 *  RND (1)
680  COLOR= 9
690  IF  RND (1) > 0.7 THEN  COLOR= 2: REM   DANGEROUS KLINGON
700  GOSUB 790: REM  PLOT NEW KLINGON POSITION
710  RETURN
720  REM  : POSITION THE KLINGONS -----
730  COLOR= 9: REM  RED
740  FOR N = 1 TO 3
750 R(N) = 34 *  RND (1):S(N) = 35 *  RND (1)
760  GOSUB 790
770  NEXT N
780  RETURN
790  REM  PLOT KLINGONS -----
800  HLIN R(N) + 1,R(N) + 3 AT S(N)
810  HLIN R(N),R(N) + 4 AT S(N) + 1
820  PLOT R(N) + 2,S(N) + 2
830  RETURN
840  REM  PLOT SPACE MINES -----
850  FOR N = 1 TO 10
860  COLOR= 2
870 A(N) = 35 *  RND (1):B(N) = 35 *  RND (1)
880  PLOT A(N),B(N)
890  NEXT N
900  RETURN
910  REM  HIT A SPACE MINE -----
920  FOR M = 1 TO 20
930 Q =  PEEK ( - 16336) +  PEEK ( - 16336)
940  NEXT M
950 WIN(P) =  INT ((WIN(P)) / 2)
960  RETURN
970  REM : HIT A KLINGON -----
980  CALL  - 1059
990 WIN(P) = WIN(P) + 1
1000  IF WIN(P) > 15 THEN 1280
1010  RETURN
1020  REM  INSTRUCTIONS -----
1030  HOME
1040  PRINT "   *****     SPACE ZAPPER     *****"
1050  PRINT "***************************************"
1060  PRINT
1070  PRINT "STEER WITH YOUR PADDLE TO HIT THE"
1080  PRINT "INVADING RED KLINGON SAUCERS."
```

```
1100  PRINT "YOU SCORE A POINT EACH TIME YOU HIT A"
1110  PRINT "KLINGON.  HOWEVER, SOMETIMES A KLINGON"
1120  PRINT "GOES CRITICAL AND TURNS BLUE."
1130  PRINT "IT IS THEN EXTREMELY DANGEROUS."
1140  PRINT "HIT A BLUE KLINGON, OR A BLUE FLOATING"
1150  PRINT "SPACE MINE, AND YOU WILL BLOW HALF YOUR SCORE"
1160  PRINT "DON'T WORRY ABOUT HITTING THE OTHER"
1170  PRINT "PLAYER."
1180  PRINT "YOU WILL FLY STRAIGHT THROUGH EACH OTHER"
1190  PRINT "FIRST PLAYER TO WIN 15 POINTS IS THE"
1200  PRINT "WINNER."
1210  PRINT
1220  INVERSE
1230  PRINT "HIT SPACE BAR TO BEGIN. "
1240  NORMAL
1250  GET K$
1260  HOME
1270  RETURN
1280  REM GAME ENDS-----
1290  HOME : VTAB 21
1300  CALL  - 1059: CALL  - 1059: CALL  - 1059
1310  FLASH : PRINT "GAME OVER !!!": NORMAL
1320  FOR J = 0 TO PMAX: PRINT "PLAYER ";J;" SCORED ";WIN(J);": ";: NEXT
    J
1330  IF WIN(0) = WIN(1) THEN  GOSUB 1500: GOTO 1360
1340 PW = (WIN(1) > WIN(0))
1350  PRINT : PRINT "PLAYER ";PW;" WINS !"
1360  PRINT "WANT ANOTHER GAME (Y/N )";
1370  GET Y$
1380  IF Y$ = "Y" THEN 200: REM  ANOTHER GAME
1390  TEXT : HOME : PRINT "HIT 'ESC' TO EXIT"
1400  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
1410  GET G$
1420  IF G$ =  CHR$ (27) THEN  HOME : PRINT "BYE. THANKS FOR PLAYING.": END

1430  PRINT : PRINT  CHR$ (4);"RUN HELLO"
1440  END
1500  PRINT "  THE GAME IS A DRAW !": RETURN
```

美式骰戲 1 (Craps)



古典賭博遊戲

這是一種大家所熟知的骰子遊戲，可供兩個人在低解析度螢光幕上遊戲。每一位遊戲者搖動兩粒骰子，如果他擲出 7 或 11 點，他就無條件地贏得一把；如果他擲出 4、5、6、8、9 或 10 點，而稍後再擲時在擲出 7 點之前若擲出相同點數，他也贏得那一把；如果他一開始就擲出 2、3 或 12，則無條件輸一把。遊戲者輪流擲骰子。

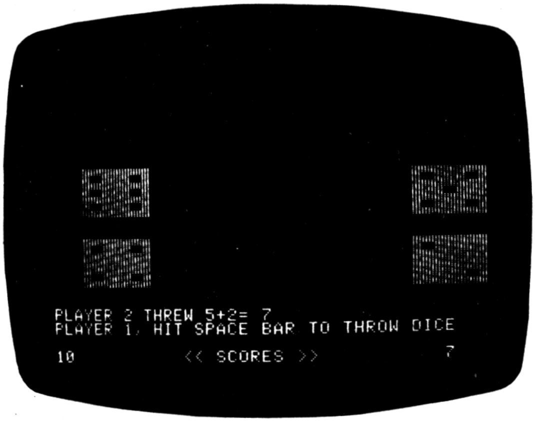
10 至 270 那幾列建立遊戲，以及產生模擬擲骰子所需的亂數。實際執行描繪骰子的低解析度副程式在100 以下那幾列。270 以下那幾列根據規則檢查骰子的點數（7 點＝贏，相同＝贏，2＝輸，……等），印出結果，並指示遊戲者下一步怎麼做。

Craps

```
10  REM CRAPS-----
20  TEXT : HOME : HTAB 12: PRINT "***   CRAPS   ***"
30  VTAB 10: HTAB 8: PRINT "WANT INSTRUCTIONS (Y/N) ?";
40  GET Y$
50  IF Y$ = "Y" THEN  GOSUB 570: REM INSTRUCTIONS
60  GR
70 P = 1: REM   PLAYER
80 DOTS = 0
90  DIM D%(2),SVE(25),TURN(2)
100 TURN(1) = 0:TURN(2) = 0
110  REM   MAIN LOOP -----
120  REM
130  FOR THROW = 1 TO 4
140 FIELD = 15 *  RND (1) + 1
150  FOR J = 1 TO 2: REM   TWO DICE
160 D%(J) = 6 *  RND (1) + 1: REM THROW DICE
170 CLICK =  PEEK ( - 16336) -  PEEK ( - 16336)
180  COLOR= FIELD
190 Y = 10 + 10 * J
200 X = 30 * P - 27
210  FOR H = X TO X + 6: VLIN Y,Y + 6 AT H: NEXT H
220  COLOR= DOTS
230  ON D%(J) GOSUB 1010,1030,1050,1070,1100,1120
240  NEXT J
250 A = D%(4 - J):B = D%(J - 1)
260  REM
270  NEXT THROW
280 TURN(P) = TURN(P) + 1
290 SVE(TURN(P)) = A + B
300  VTAB 21: PRINT "PLAYER ";P;" THREW ";A;"+";B;"= ";A + B
310  IF TURN(P) > 1 THEN  GOTO 350
320  IF A + B = 2 OR A + B = 3 OR A + B = 12 THEN  GOTO 400
330  IF A + B = 7 OR A + B = 11 THEN  GOTO 490
340  GOTO 420
350  IF A + B = 7 THEN  GOTO 490
360  PRINT "THROWS SO FAR: ";
370  FOR Q = 1 TO TURN(P): PRINT SVE(Q);", ";: NEXT Q: PRINT
380  FOR Q = 1 TO TURN(P) - 1: IF A + B = SVE(Q) THEN  GOTO 470: NEXT Q
390  GOTO 420
400  CALL  - 1059: CALL  - 1059: PRINT "SORRY, YOU LOSE--";: GOTO 520
410  REM PROMPT NEXT PLAYER-----
420 P = 3 - P: REM  CHANGE PLAYERS
430  VTAB 23: PRINT "PLAYER ";P;", HIT SPACE BAR TO THROW DICE"
440  GET G$
450  IF G$ = "P" THEN  GOTO 420
460  GOTO 130
470  REM WIN BY MATCHING-----
480  VTAB 22: HTAB 23: PRINT "--MATCHING WIN!!"
```

```
490   REM WINNER-----
500   FOR M = 1 TO 3: GOSUB 730: FOR PAUSE = 1 TO 50: NEXT PAUSE: NEXT M: REM
        TRIPLE BUZZ
510   VTAB 23: PRINT "YOU WIN!! ";
520   PRINT "WANT ANOTHER GAME (Y/N)     "
530   GET Y$
540   IF Y$ <  > "Y" THEN  TEXT : HOME : PRINT "BYE.": END
550   GR : HOME
560 P = 1:TURN(1) = 0:TURN(2) = 0: GOTO 430: REM  NEW GAME
570   REM INSTRUCTIONS-----
580   HOME : VTAB 5
590   PRINT "IN CRAPS YOU THROW TWO DICE.  A 7 OR 11"
600   PRINT "THROW WINS."
610   PRINT "YOU CAN  ALSO WIN BY THROWING A 4,5,6,"
620   PRINT "8,9,10 AND  MATCHING IT BEFORE"
630   PRINT "THROWING A 7."
640   PRINT "IF ON THE FIRST THROW A 2,3 OR 12 COMES"
650   PRINT "UP, YOU LOSE AUTOMATICALLY."
660   PRINT : PRINT
670   HTAB 8: PRINT "***   GOOD LUCK !!!   ***"
680   PRINT : PRINT
690   PRINT "PLAYER #1, HIT SPACE BAR FOR YOUR FIRST"
700   PRINT "ROLL."
710   GET G$
720   RETURN
730   REM  BUZZ-----
740   FOR M = 1 TO 20:BUZZ =  PEEK ( - 16336) -  PEEK ( - 16336): NEXT M: RETURN
1000  REM PLOT DICE-----
1010  REM ONE---
1020  PLOT X + 3,Y + 3: RETURN
1030  REM TWO---
1040  PLOT X + 1,Y + 1: PLOT X + 5,Y + 5: RETURN
1050  REM THREE---
1060  GOSUB 1010: GOSUB 1030: RETURN
1070  REM FOUR---
1080  GOSUB 1030
1090  PLOT X + 5,Y + 1: PLOT X + 1,Y + 5: RETURN
1100  REM FIVE---
1110  GOSUB 1070: GOSUB 1010: RETURN
1120  REM SIX---
1130  GOSUB 1070
1140  PLOT X + 1,Y + 3: PLOT X + 5,Y + 3: RETURN
```

美式骰戲 2 (Diddle)



另一種骰子遊戲

這是另外一種與前面的Craps（美式骰戲1）完全不同的骰子遊戲；在這個遊戲裡，誰最先累積到 30 點誰就是贏家。如果有一粒骰子出現1點，那一把將不計分，但是如果兩粒骰子都出現1點（雙么），您將會失去整個分數！然而，如果您擲出其他相同點數的組合（雙二至雙六），您將免費多擲一把。

美式骰戲2（Diddle）與美式骰戲1（Craps）的程式有許多地方相同，您可能注意到在這兩個遊戲中 10至270那幾列（建立遊戲和擲骰子）與1000以下那幾列（描繪骰子）實際上是大致相同的。這有兩項優點：其一爲只要您打好Craps程式，您就等於完成Diddle程式的一半：

```
LOAD CRAPS (assuming you already have it on
diskette)
DEL 275, 990
```

您只要對 10至270那幾列稍做修改，然後再打入Diddle 的中間部份；其二爲如果您想自己設計骰子遊戲，自己訂規則，您也只要修改中間部份即可。您可以

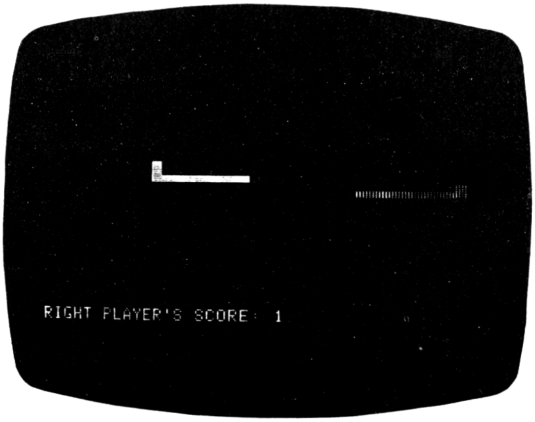
試試看！

Diddle

```
10   REM DIDDLE -----
20   TEXT : HOME : HTAB 11: PRINT "*** DIDDLE   ***"
30   VTAB 10: HTAB 8: PRINT "WANT INSTRUCTIONS (Y/N ?";
40   GET Y$
50   IF Y$ = "Y" THEN  GOSUB 670: REM INSTRUCTIONS
60   TEXT : HOME : GR
70 P = 1: REM   PLAYER
80 DOTS = 0
90 BLANK$ = "                    ": REM  20 BLANKS
100 SCRE(1) = 0:SCRE(2) = 0: REM PLAYERS' SCORES
110  REM   MAIN LOOP -----
120  VTAB 21: PRINT "PLAYER ";P;" THREW ";A;"+";B;"= ";A + B
130  FOR THROW = 1 TO 5 + 5 *  RND (1)
140 FIELD = 15 *  RND (1) + 1
150  FOR J = 1 TO 2
160 D%(J) = 6 *  RND (1) + 1: REM THROW DICE
170 CLICK =  PEEK ( - 16336) -  PEEK ( - 16336)
180 Y = 10 + 10 * J
190 X = 30 * P - 27
200  COLOR= FIELD
210  FOR H = X TO X + 6: VLIN Y,Y + 6 AT H: NEXT H
220  COLOR= DOTS
230  ON D%(J) GOSUB 1010,1030,1050,1070,1100,1120
240  NEXT J
250 A = D%(4 - J):B = D%(J - 1)
260  VTAB 21: HTAB 8: PRINT P;: HTAB 16: PRINT A;: HTAB 18: PRINT B;: HTAB
   21: PRINT A + B;"  "
270  NEXT THROW
280 SCRE(P) = SCRE(P) + A + B
290  IF A = 1 OR B = 1 AND A <  > B THEN  GOSUB 440
300  IF A = B THEN  GOSUB 490
310  GOTO 340
320  IF A = 1 AND B = 1 THEN  GOTO 340
330  NEXT H
340  VTAB 24: PRINT SCRE(1); SPC( 11);"<< SCORES >>"; SPC( 11);SCRE(2);
350  FOR K = 1 TO 2: IF SCRE(K) > 30 THEN 600: NEXT K: REM GAME OVER
360 P = 3 - P: REM CHANGE PLAYERS
370  REM  PROMPT NEXT PLAYER-----
380  VTAB 22: HTAB 1: PRINT "PLAYER ";P;", HIT SPACE BAR TO THROW DICE"
390  GET G$
400  VTAB 22: PRINT BLANK$
```

```
410  IF G$ = "P" THEN 340
420  VTAB 21: HTAB 20: PRINT BLANK$ + BLANK$ + BLANK$
430  GOTO 130
440  REM THROW A ONE-----
450 SUCCESS(P) = SUCCESS(P) - A - B
460  VTAB 21: HTAB 20: PRINT "NO SCORE THIS TURN"
470  GOSUB 820
480  RETURN
490  REM THROW A DOUBLE-----
500  VTAB 21: HTAB 20: PRINT "DOUBLE ";A;"--";
510  IF A = 1 THEN  GOTO 560
520  GOSUB 820: GOSUB 820: GOSUB 820: GOSUB 820
530  IF A <  > 1 THEN  PRINT "FREE TURN."
540 P = 3 - P: REM CHANGE PLAYERS
550  RETURN
560  REM THROW DOUBLE ONES-----
570  PRINT "SORRY, YOU JUST LOST YOUR ENTIRE SCORE !!!"
580 SUCCESS(P) = 0
590  GOTO 550
600  REM WINNER
610  FOR J = 1 TO 3: GOSUB 820: NEXT J
620  VTAB 21: HTAB 1: PRINT "PLAYER ";K;" IS THE WINNER !!"
630  VTAB 22: PRINT "WANT ANOTHER GAME (Y/N) ": GET Y$
640  HOME
650  IF Y$ <  > "Y" THEN  HOME : PRINT "BYE": END
660 P = 1:SUCCESS(1) = 0:SUCCESS(2) = 0: GOTO 380: REM  NEW GAME
670  REM INSTRUCTIONS-----
680  HOME
690  PRINT "DIDDLE IS A GAME OF DICE.  OBJECT IS TO"
700  PRINT "SCORE 30.  ON EACH TURN, HIT SPACE BAR"
710  PRINT "TO THROW YOUR TWO DICE, OR HIT 'P'TO"
720  PRINT "PASS.  IF YOU THROW A ONE YOU LOSE YOUR"
730  PRINT "SCORE FOR THAT TURN ONLY.   DOUBLE ONE"
740  PRINT "LOSES YOUR ENTIRE SCORE !  ANY OTHER"
750  PRINT "DOUBLE GETS YOU AN EXTRA FREE TURN."
760  PRINT : PRINT
770  PRINT "   ***   GOOD LUCK   !!!   ***"
780  PRINT : PRINT
790  PRINT "PLAYER #1, HIT SPACE BAR FOR YOUR FIRST ROLL."
800  GET G$
810  RETURN
820  REM BUZZ-----
830  FOR M = 1 TO 20:BUZZ =  PEEK ( - 16336) -  PEEK ( - 16336): NEXT M: RETURN
1000  REM PLOT DICE-----
1010  REM ONE---
1020  PLOT X + 3,Y + 3: RETURN
1030  REM TWO---
1040  PLOT X + 1,Y + 1: PLOT X + 5,Y + 5: RETURN
1050  REM THREE---
1060  GOSUB 1010: GOSUB 1030: RETURN
1070  REM FOUR---
1080  GOSUB 1030
1090  PLOT X + 5,Y + 1: PLOT X + 1,Y + 5: RETURN
1100  REM FIVE---
1110  GOSUB 1070: GOSUB 1010: RETURN
1120  REM SIX---
1130  GOSUB 1070
1140  PLOT X + 1,Y + 3: PLOT X + 5,Y + 3: RETURN
```

踩尾巴(Cross Porpoises)



儘力去踩對手的尾巴，但小心不要被別人踩到！

在這個遊戲裡，每位遊戲者各有一條以低解析圖形在螢光幕上到處爬行的長尾小毛蟲，每位遊戲者要操縱各人的小毛蟲上、下、左、右移動去儘力踩別人的尾巴，也就是用頭連接別人的尾巴，每踩一次獲得一分。當然，您也要提防別人踩到您的尾巴。

每位遊戲者都坐在鍵盤前面，左邊遊戲者分別利用E、X、S和D鍵控制他的小毛蟲上、下、左、右移動，而右邊遊戲者利用I、M、J和K做同樣的事情——您可能已經很熟悉這四個鍵，因為它們可在編修（edit）程式時用來移動游標；這個副程式已在第一章說明過。一旦您按了鍵改變小毛蟲的方向後，它就會維持那個方向，一直到您再改變它為止。如果它爬抵螢光幕的邊緣，它會回頭以相反方向繼續爬行，或者就停在原地一直到您下達新指示……小心您的對手當時躲在您的身邊！您不久就會發現您可以在鍵盤上利用手指輕快地來回跳彈，讓您的小毛蟲以極快的速度左扭右歪或打轉以逃避對手的緊迫盯人，但是您的對手

仍會傾全力咬住您。每一次您被踩到時您都會聽到一個聲響——短聲表示左邊遊戲者得分，長聲表示右邊的遊戲者得分。

開始時的小毛蟲——紅的給左邊遊戲者，白的給右邊遊戲者——是在140至160那幾列畫在螢光幕的中間。180至220那幾列檢查那些鍵被按了，並據以改變小毛蟲的方向，280至300那幾列檢查是否有小毛蟲太靠近螢光幕邊緣，而270那一列檢查您是否成功地攔截了您的對手，這是藉SCRN功能檢查您的小毛蟲前面的小方塊顏色而完成的；如果檢查的結果為正，370至390那幾列會產生一個聲響，並將勝利遊戲者的分數加1，再將分數PRINT 在螢光幕的底部。310至340那幾列描繪小毛蟲的現有位置，並將它們的舊位置擦去。

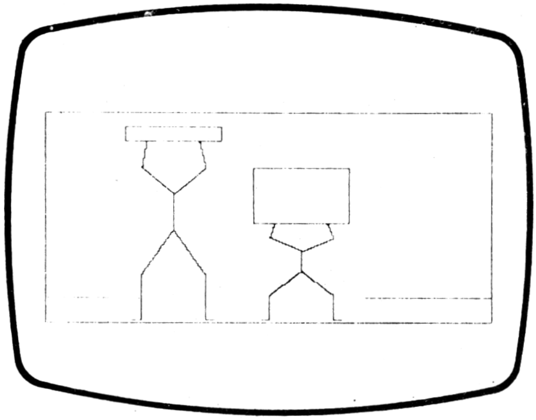
P代表遊戲者（Player）。在這個遊戲裡，每移動一步都是在“FOR P＝0 TO 1 …… NEXT P”迴圈中完成的，因此每經由迴圈一次，兩位遊戲者都各被考慮一次。變數X（P, NOW）和Y（P, NOW）分別代表0號（左邊）遊戲者和1號（右邊）遊戲者所屬小毛蟲現在（NOW）的位置。

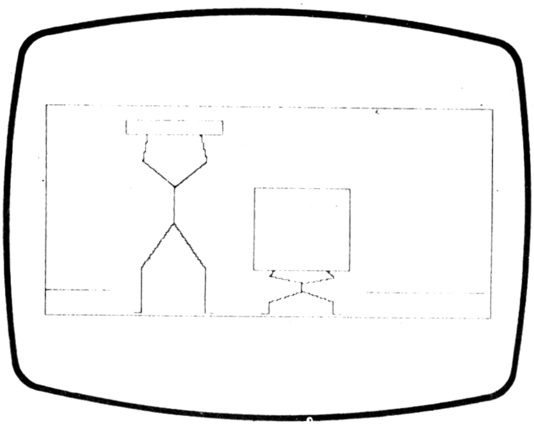
Cross Porpoises

```
10  REM CROSS PORPOISES---
20  DIM X(2,300),Y(2,300)
30 NOW = 10
40 C(0) = 1:C(1) = 14: REM COLORS FOR PLAYERS
50  TEXT : HOME : VTAB 10
60  PRINT "WANT INSTRUCTIONS (Y/N) ?"
70  GET Y$
80  IF Y$ = "Y" THEN  GOSUB 410
90  REM PLOT BORDER---
100  GR
```

```
110  COLOR= 3: HLIN 1,38 AT 1: HLIN 1,38 AT 38: VLIN 1,38 AT 1: VLIN 1,38
      AT 38
120  REM FILL ARRAY TO PLOT INITIAL PORPOISES---
130  FOR P = 0 TO 1
140  COLOR= C(P)
150  FOR J = 1 TO NOW:X(P,J) = 15 + 10 * P:Y(P,J) = 5 + J: PLOT X(P,J),Y(
     P,J): NEXT J
160  NEXT P
170  REM KEYPRESSES CONTROL MOTION---
180 KEY =  PEEK ( - 16384) - 128
190 XSPD(0) = (KEY = 68) - (KEY = 83): REM D,S
200 YSPD(0) = (KEY = 88) - (KEY = 69): REM X,E
210 XSPD(1) = (KEY = 75) - (KEY = 74): REM K,J
220 YSPD(1) = (KEY = 77) - (KEY = 73): REM  M,I
230 NOW = NOW + 1
240  IF NOW > 295 THEN  GOTO 30
250  REM  PLOT PORPOISES---
260  FOR P = 0 TO 1
270  IF  SCRN( X(P,NOW),Y(P,NOW)) = C(1 - P) THEN  GOSUB 360: REM  HIT OP
     PONENT---
280 X(P,NOW) = X(P,NOW - 1) + XSPD(P)
290 Y(P,NOW) = Y(P,NOW - 1) + YSPD(P)
300  IF  SCRN( X(P,NOW),Y(P,NOW)) = 3 THEN XSPD(P) =  - 5 * XSPD(P):YSPD(
     P) =  - 5 * YSPD(P): GOTO 280: REM AVOID EDGE---
310  COLOR= 15: PLOT X(P,NOW),Y(P,NOW): REM PLOT HEAD
320  COLOR= C(P): PLOT X(P,NOW - 1),Y(P,NOW - 1): REM PLOT NECK
330  COLOR= 0: PLOT X(P,NOW - 10),Y(P,NOW - 10): REM  ERASE TAIL
340  NEXT P
350  GOTO 180
360  REM HIT OPPONENT---
370  CALL  - 1059: CALL  - 1059: REM BEEP
380 HIT(P) = HIT(P) + 1
390  VTAB 21 + 2 * P: PRINT  MID$ ("LEFT RIGHT",5 * P + 1,5);" PLAYER'S S
     CORE: ";HIT(P)
400  RETURN
410  REM INSTRUCTIONS---
420  HOME : VTAB 5
430  PRINT "       ***   CROSS PORPOISES    ***"
440  PRINT : PRINT "A GRAPHICS GAME FOR 2 PLAYERS."
450  PRINT : PRINT "LEFT PLAYER MOVES HIS PORPOISE UP, DOWN"
460  PRINT "LEFT, RIGHT BY HITTING E,X,S,D."
470  PRINT : PRINT "RIGHT PLAYER MOVES HIS PORPOISE UP, DOWN";
480  PRINT "LEFT, RIGHT BY HITTING I,M,J,K."
490  PRINT : PRINT "BUTT OPPONENT'S PORPOISE IN THE SIDE TO"
500  PRINT "SCORE POINTS."
510  PRINT : PRINT "HIT ANY KEY TO START..."
520  VTAB 21: PRINT " E "," I "
530  VTAB 22: PRINT "S D","J K"
540  VTAB 23: PRINT " X "," M "
550  GET Y$
560  RETURN
```

舉重選手(Weight Lifters)





如果您的控制器轉得不夠，您將會被地心引力壓平

這是一個供兩人玩的高解析度動作遊戲，適合各年級兒童遊戲。螢光幕上描繪出兩個人形圖樣，每一個人的頭上都舉著一樣重物。正如您看到的，情況是很危險的：兩位舉重選手不斷地被畫在螢光幕上，但地心引力逼使他們當著您的眼前向下縮小，當重量愈來愈重時它們也愈縮愈小。解救的方法就在您手上，但只有一種：每位遊戲者必須儘快地來回旋轉他的控制器去解救他的舉重選手使它停止縮小，不得停頓——這是一個動作遊戲，愈快的人贏！動作慢的人將會懊喪地

看著他的舉重選手漸漸地沒入地下；當一方的舉重選手沒入地下時比賽就結束了。

描繪人形的繪圖副程式在 610 至 660 那幾列中，注意其中的 K、W、S、E 和 H 各變數，他們分別代表膝（Knee），腰（Waist），肩（Shoulder），肘（Elbow）和手（Hand）的高度，這些高度在 310 至 370 那幾列中不斷地重新設定。旋轉控制器較慢的人將會看到他的舉重選手穩定地縮小，而旋轉較快的舉重選手則維持同樣高度。380 至 460 那幾列讀取控制器的讀數，並檢查那一個人轉得比較快；其方法在第一章中討論過。

小孩是很喜歡玩這遊戲的，但要小心！他們很可能磨壞您的控制器。

Weight Lifters

```
100  REM WEIGHT LIFTERS-----
110  REM
120  TEXT : HOME : VTAB 10
130  PRINT "WANT INSTRUCTIONS (Y/N) ?"
140  GET Y$
150  IF Y$ = "Y" THEN  GOSUB 810
160  REM SET VARIABLES-----
170 YY = 157: REM POSITION OF FEET
180 WD = 20: REM  WIDTH
190  HOME : VTAB 21: PRINT "       PLAYER #0   PLAYER #1"
200  FOR J = 0 TO 1
210 HT(J) = 150: REM  HEIGHT
220 M(J) = 80 + 80 * J: REM  POSITION OF MAN
230  GOSUB 310
240  NEXT J
250  HGR : HCOLOR= 7
260  HPLOT 0,0 TO 279,0 TO 279,159 TO 0,159 TO 0,0
270  HPLOT 0,140 TO 40,140
280  HPLOT 200,140 TO 279,140
290 J = 0
300  GOTO 670: REM MAIN LOOP
```

```
310  REM  SET SIZE OF FIGURE -----
320 K = YY - .25 * HT(J): REM  KNEE
330 W = YY - .5 * HT(J): REM    WAIST
340 S = YY - .70 * HT(J): REM   SHOULDER
350 E = YY - .85 * HT(J): REM   ELBOW
360 H = YY - HT(J): REM   HAND
370  RETURN
380  REM  READ PADDLE -----
390  FOR N = 1 TO 2
400 PAD(J) =  PDL (J)
410  FOR T = 1 TO 25: NEXT T
420 TRY(J) =  ABS ( PDL (J) - PAD(J))
430  IF TRY(J) > TRY(1 - J) THEN HT(J) = HT(J) * 1.1
440  NEXT N
450  IF HT(J) > 159 THEN HT(J) = 159
460  RETURN
470  REM  GAME OVER -----
480  CALL  - 1059: CALL  - 1059
490  VTAB 22: HTAB 1
500  PRINT "PLAYER # ";1 - J;" HAS WON !!!"
510  PRINT "WANT ANOTHER GAME (Y/N) ?"
520  GET Y$
530  IF Y$ = "Y" THEN  HOME : GOTO 160
540  TEXT : HOME : VTAB 10
550  PRINT "HIT 'ESC' TO EXIT"
560  PRINT "HIT ANY OTHER KEY TO RETURN TO ME NU"
570  GET H$
580  IF H$ =  CHR$ (27) THEN  HOME : VTAB 5: PRINT "OK. THANKS FOR PLAYIN
     G.": END
590  PRINT : PRINT  CHR$ (4);"RUN HELLO"
600  END
610  REM   PLOT FIGURE-----
620  HPLOT M - 1.3 * WD,YY TO M - WD,YY TO M - WD,K TO M,W TO M + WD,K TO
     M + WD,YY TO M + 1.3 * WD,YY
630  HPLOT M - .8 * WD,H TO M - WD,E TO M,S TO M + WD,E TO M + .8 * WD,H
640  HPLOT M,S TO M,W
650  HPLOT M - 1.5 * WD,H TO M + 1.5 * WD,H TO M + 1.5 * WD,H / 2 TO M -
     1.5 * WD,H / 2 TO M - 1.5 * WD,H
660  RETURN
670  REM  NEW MAIN LOOP-----
680 J = 1 - J
690  IF HT(J) < 20 THEN 470: REM  GAME OVER
700 HT = HT(J):M = M(J)
710  HCOLOR= 0
720  GOSUB 310
730  GOSUB 380: REM   READ PADDLE
740  GOSUB 610: REM  ERASE OLD FIGURE
750 HT(J) = .90 * HT(J): REM GRAVITY SHRINKS HEIGHT
760  HCOLOR= 7
770 HT = HT(J)
780  GOSUB 310
790  GOSUB 610: REM   DRAW NEW FIGURE
800  GOTO 670
810  REM  INSTRUCTIONS-----
820  HOME : VTAB 2: HTAB 15
830  PRINT "WEIGHT LIFTERS"
840  PRINT : FOR M = 1 TO 8: PRINT "*****";: NEXT M
850  PRINT
860  PRINT "WEIGHT LIFTERS IS AN ACTION GAME FOR TWO PLAYERS USING THE PA
     DDLES"
870  PRINT : PRINT "YOUR WEIGHT LIFTER WILL START TO DROOP"
880  PRINT "UNDER HIS WEIGHT, UNLESS YOU HOLD HIM UP BY TURNING YOUR PADD
     LE TO AND FRO"
890  PRINT "REALLY FAST -- FASTER THAN THE OTHER"
900  PRINT "PLAYER!!"
910  PRINT : PRINT "THE FIRST PLAYER TO GET PUSHED DOWN TO"
920  PRINT "THE HORIZONTAL LINE IS THE LOSER."
930  PRINT
940  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE...   ";: NORMAL
950  GET H$
960  RETURN
```



浪子回頭(Zig-Zag)

這是一個可供兩人在一張大小於遊戲開始時選定的方格圖上遊玩的策略遊戲，每一邊的方格數必須爲奇數，如5×7或9×9。遊戲從中央的方格開始，每一位遊戲者必須藉"U"(上)，"D"(下)，"L"(左)或"R"(右)諸鍵輪流依次加上格子，如此可在螢光幕上產生一條由中心開始到處蛇行的折線，但是每人只能往蛇頭方向前進。每位遊戲者的目標是設法使Z字形折線回到各

人的"家"：左上角的小方塊給藍方（ Blue ），右下角的小方塊給黃方（Yellow ），誰走最後一步，或被迫走最後一步到家都有關係。

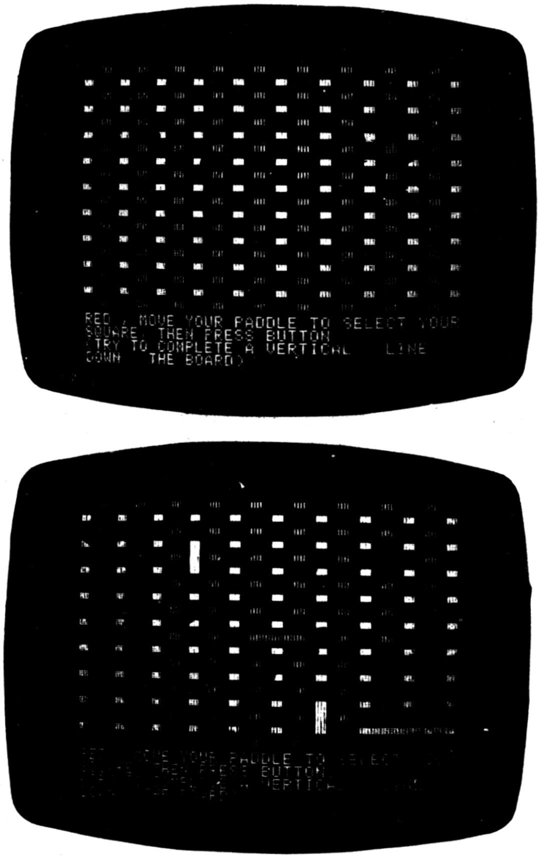
提示：說也奇怪，先走的人往往只要朝離開他家的方向起步就可以做得很好。每位遊戲者應該儘量避免走到板子的邊緣，因爲這樣可讓對方隨便轉向一邊而佔了便宜。

Zig-Zag

```
3 DOG
10  REM *******************
20  REM *                 *
30  REM *     ZIGZAG      *
40  REM *                 *
50  REM *       BY        *
60  REM *                 *
70  REM *  STUART ANSTIS  *
80  REM *                 *
90  REM *******************
100  REM
110  HOME : VTAB 10: PRINT "WANT INSTRUCTIONS (Y/N) ?"
120  GET Y$
130  IF Y$ = "Y" THEN  GOSUB 960
140 P$ = "YELLOW  BLUE"
150  TEXT : HOME : VTAB 10
160  PRINT "TYPE IN THE WIDTH AND HEIGHT OF GRID YOUWANT."
170  VTAB 15: HTAB 1
180  INPUT "HOW WIDE (3, 5, 7, 9)  ";WIDE
190  IF (WIDE - 3) * (WIDE - 5) * (WIDE - 7) * (WIDE - 9) <  > 0 THEN  CALL
   - 1059: GOTO 180
200  INPUT "HOW HIGH (3, 5, 7, 9)  ";HI
210  IF (HIGH - 3) * (HIGH - 5) * (HIGH - 7) * (HIGH - 9) <  > 0 THEN  CALL
   - 1059: GOTO 200
220  REM  PLOT GRID -----
230  GR : COLOR= 15
240  FOR I = 0 TO 4 * WIDE STEP 4
250  VLIN 0,4 * HIGH AT I
260  NEXT I
270  FOR I = 0 TO 4 * HIGH STEP 4
280  HLIN 0,4 * WIDE AT I
290  NEXT I
300  COLOR= 2:X = 2:Y = 2
310  HLIN X - 1,X + 1 AT Y: VLIN Y - 1,Y + 1 AT X
320  COLOR= 13:X = 4 * WIDE - 2:Y = 4 * HIGH - 2
330  HLIN X - 1,X + 1 AT Y: VLIN Y - 1,Y + 1 AT X
340 X = 2 * WIDE:Y = 2 * HIGH
350  HOME : VTAB 21
360  PRINT  MID$ (P$,1 + 6 * P,6);"'S TURN:--"
370  PRINT "UP, DOWN, LEFT, RIGHT ?"
375  PRINT "(U,D,L,R)"
380  REM  MAIN LOOP-----
390 HUE = 2 + 11 * P
400  FOR C = HUE TO 0 STEP  - HUE
410  COLOR= C
420  GOSUB 620
430 KEY =  PEEK ( - 16384)
440  IF C <  > 0 AND KEY > 127 THEN  GOTO 470
450  NEXT C
460  GOTO 380
470  POKE  - 16368,0: REM   CLEAR KEYBOARD
480 H$ =  CHR$ (KEY - 128)
490  HOME : VTAB 21
500  PRINT  MID$ (P$,1 + 6 * (1 - P),6);"'S TURN:--"
510  PRINT "UP, DOWN, LEFT, RIGHT ?"
515  PRINT "(U,D,L,R)"
520  IF H$ <  > "U" AND H$ <  > "D" AND H$ <  > "L" AND H$ <  > "R" THEN
   CALL  - 1059: PRINT "YOU MUST HAVE HIT THE WRONG KEY: TRY": PRINT "
  AGAIN": GOTO 400
530  IF H$ = "U" THEN V = Y - 4:H = X: GOTO 670
540  IF H$ = "D" THEN V = Y + 4:H = X: GOTO 670
550  IF H$ = "L" THEN H = X - 4:V = Y: GOTO 670
560  IF H$ = "R" THEN H = X + 4:V = Y: GOTO 670
570 X = H:Y = V
580  IF X = 2 AND Y = 2 THEN  GOSUB 810: REM  GAME OVER
590  IF X = 4 * WIDE - 2 AND Y = 4 * HIGH - 2 THEN  GOSUB 810: REM   GAME
   OVER
600  GOTO 380
610  END
```

```
620  REM  PAINT SQUARE -----
630  FOR L =  - 1 TO  + 1
640  HLIN X - 1,X + 1 AT Y + L
650  NEXT L
660  RETURN
670  REM  CHECK LEGALITY OF MOVE-----
680  IF H < 2 OR H > 4 * WIDE OR V < 2 OR V > 4 * HIGH THEN  GOTO 770
690  IF  SCRN( H + 1,V + 1) <  > 0 THEN  GOTO 720
700 P = 1 - P: REM  NEXT PLAYER
710  GOTO 570
720  REM  SQUARE ALREADY FILLED-----
730  CALL  - 1059
740  HOME : VTAB 21: PRINT "SORRY, THAT SQUARE IS ALREADY FILLED:"
750  PRINT "TRY AGAIN"
760  GOTO 380
770  REM  PROPOSED MOVE OFF EDGE OF BOARD-----
780  CALL  - 1059
790  HOME : VTAB 21: PRINT "SORRY, THAT IS OFF THE EDGE OF THE BOARD: TRY
      AGAIN"
800  GOTO 380
810  REM  GAME OVER-----
820  FOR J = 1 TO 5: CALL  - 1059: NEXT J
830  HOME : VTAB 21
840  IF X = 2 THEN  PRINT "BLUE HAS WON !!!": GOTO 860
850  IF X <  > 2 THEN  PRINT "YELLOW HAS WON!!": GOTO 860
860  PRINT "WANT ANOTHER GAME (Y/N) ?"
870  GET Y$
880  IF Y$ = "Y" THEN  GOTO 150
890  TEXT : HOME : VTAB 10
900  PRINT "HIT'ESC' TO EXIT"
910  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
920  GET H$
930  IF H$ =  CHR$ (27) THEN  HOME : VTAB 5: PRINT "OK. THANKS FOR PLAYIN
     G.": END
940  PRINT : PRINT  CHR$ (4);"RUN HELLO"
950  END
960  REM  INSTRUCTIONS-----
970  HOME : VTAB 2: HTAB 18
980  PRINT "ZIGZAG"
990  PRINT
1000  FOR M = 1 TO 5: PRINT "********";: NEXT M
1010  PRINT
1020  PRINT "ZIGZAG IS A GAME OF STRATEGY FOR TWO": PRINT "PLAYERS: YELLO
     W AND BLUE."
1030  PRINT : PRINT "YOU WILL SPECIFY THE SIZE OF THE GRID ON WHICH YOU W
     ILL PLAY."
1040  PRINT "EACH PLAYER IN TURN ADDS A SQUARE, UP,"
1050  PRINT "DOWN, LEFT OR RIGHT FROM THE PREVIOUS"
1060  PRINT "SQUARE. THE SQUARES FORM A ZIGZAG LINE"
1070  PRINT "OVER THE BOARD. BLUE'S OBJECTIVE IS TO"
1080  PRINT "MAKE THE LINE REACH THE TOP LEFT CORNER."
1090  PRINT "YELLOW'S OBJECTIVE IS TO GET THE LINE TO THE BOTTOM RIGHT CO
     RNER."
1100  PRINT : PRINT "HINT: DON'T REACH THE EDGE FIRST OR THE OTHER PLAYER
      CAN TURN THE LINE TO HIS"
1110  PRINT "ADVANTAGE."
1120  PRINT
1130  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE...  ";: NORMAL
1140  GET H$
1150  RETURN
```

搭橋(Bridge-It)



遊戲的配置與進行狀況

這是一個供兩人玩的策略遊戲，板上包含許多縱橫排列的紅藍低解析度點，每一步都要包含任意兩個相鄰點的連接。先走的人（藍方）要設法在螢光幕上建立一條由左至右的完整路徑，而第二位遊戲者（紅方）要設法連接一條由上向下的垂直路徑，每位遊戲者都不可以跨越別人的線或顏色，因此大部份策略要放在設法阻隔對方。

藍方使用 0 號遊戲控制器，紅方使用 1 號控制器。輪到任何一方時，那一方

遊戲者要藉轉動控制器來選擇點。正如大多數這類遊戲一樣，順時針轉動控制器會使一個小指標由左至右沿著第一列橫過螢光幕，然後第二列，…，像讀書一樣。當選好一個點時，遊戲者可按" H "畫一條水平線，或者按" V "畫一條垂直線連接下面同顏色的點。

Bridge-It

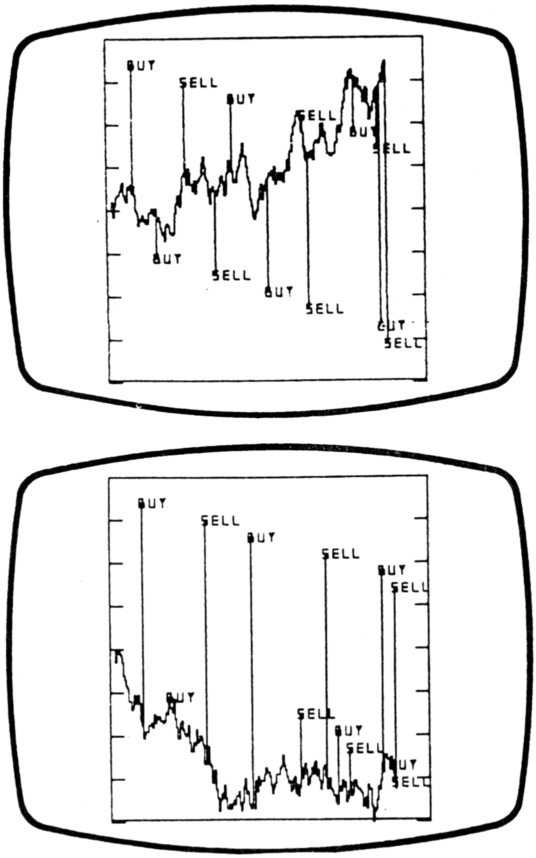
```
10   REM  ******************
20   REM  *    BRIDGE-IT   *
30   REM  *       BY       *
40   REM  * STUART ANSTIS  *
50   REM  ******************
60   REM
65   GOSUB 3100: REM  INITIAL POKES FOR TUNE
70   TEXT : HOME : VTAB 2: HTAB 18
80   PRINT "BRIDGE-IT"
90   PRINT
100   FOR M = 1 TO 40: PRINT "*";: NEXT M
110   VTAB 10
120   PRINT "WANT INSTRUCTIONS (Y/N) ?"
130   GET Y$
140   IF Y$ = "Y" THEN  GOSUB 800
150   REM  INITIALIZE VARIABLES-----
160  SIZE = 9
170  P$ = "RED BLUE"
180  D$ = "VERTICAL  HORIZONTAL"
190  E$ = "DOWN  ACROSS"
200  P = 0
210   GR
220   GOSUB 460: REM  PLOT THE BOARD
230  DIV = 256 / (SIZE * SIZE)
240   GOTO 740
250   REM  LINES 170 TO 330 ARE MAIN LOOP-----
```

```
260 PADDLE =  PDL (P) / DIV
270 POLD =  PDL (P)
280 X =  INT (SIZE * (PADDLE / SIZE -  INT (PADDLE / SIZE))): REM  PADDLE
    MOD SIZE
290 Y =  INT (PADDLE / SIZE)
300 X = 4 * X + 2 - 2 * (P = 1)
310 Y = 4 * Y + 2 + 2 * (P = 1)
320  REM  FLICKER THE SQUARE-----
330  FOR C = 0 TO 1
340  COLOR= 0
350  IF C = 1 THEN  COLOR= 1 + 6 * P
360  PLOT X,Y
370  FOR J = 1 TO 5
380 PUSH =  PEEK ( - 16287 + P)
390  IF PUSH > 127 THEN 540
400 MOVE =  PEEK ( - 16384)
410  IF MOVE > 127 THEN 600
420  NEXT J
430  NEXT C
440  GOTO 260
450  REM  PLOT THE BOARD-----
460  FOR K = 0 TO 1
470  COLOR= 7 - 6 * K
480  FOR R = 4 - 2 * K TO 39 STEP 4
490  FOR C = 2 * K TO 39 - 3 * K STEP 4
500  PLOT C,R
510  NEXT C,R
520  NEXT K
530  RETURN
540  REM  READ KEY PRESS-----
550  CALL  - 1059
560  HOME
570  VTAB 21: HTAB 1: PRINT "DO YOU WANT TO MOVE DOWN (D) OR"
580  PRINT "TO THE RIGHT (R) ?";
590  GOTO 400
600  REM  READ KEYBOARD-----
610  POKE  - 16368,0: REM  CLEAR KEYBOARD
620 M$ =  CHR$ (MOVE - 128)
630  IF M$ <  > "D" AND M$ <  > "R" THEN  HOME : VTAB 22: PRINT "SORRY, T
    RY AGAIN.": FOR T = 1 TO 500: NEXT T: GOTO 570
640 S =  SCRN( X + 2 * (M$ = "R"),Y + 2 * (M$ = "D"))
650  IF S <  > 0 THEN 690
660  IF X > 33 AND M$ = "R" THEN 690
670  IF Y > 35 AND M$ = "D" THEN 690
680  GOTO 700
690  CALL  - 1059: CALL  - 1059: PRINT : PRINT : PRINT : PRINT "SORRY, ";
    MID$ (P$,4 * P + 1,4);", THAT MOVE IS BLOCKED.  MOVE YOUR PADDLE TO
    A NEW SQUARE.": GOTO 260
700  COLOR= 1 + 6 * P
705  ON P + 1 GOSUB 3020,3030: REM  PLAY TUNE
710  IF M$ = "D" THEN  VLIN Y,Y + 4 AT X
720  IF M$ = "R" THEN  HLIN X,X + 4 AT Y
730 P = 1 - P
740  HOME : VTAB 24
750  PRINT  MID$ (P$,4 * P + 1,4);", MOVE YOUR PADDLE TO SELECT YOUR"
760  PRINT "SQUARE, THEN PRESS BUTTON."
770  PRINT "(TRY TO COMPLETE A "; MID$ (D$,1 + 10 * P,10);" LINE"
780  PRINT  MID$ (E$,1 + 6 * P,6);" THE BOARD)";
790  GOTO 260
800  REM  INSTRUCTIONS-----
810  HOME
820  PRINT "BRIDGE-IT IS A STRATEGY GAME FOR TWO"
830  PRINT "PLAYERS.  THE FIRST PLAYER (RED) MOVES"
840  PRINT "PADDLE #0 TO SELECT A DOT. (THE"
850  PRINT "PADDLE MOVES THE POINTER ALONG THE"
860  PRINT "FIRST ROW, THEN ALONG THE SECOND, LIKE"
870  PRINT "READING A BOOK)."
880  PRINT : PRINT "WHEN THE PLAYER HAS CHOSEN HIS DOT,"
890  PRINT "HE PRESSES HIS PUSHBUTTON. HE THEN HITS"
900  PRINT "'R' TO CONNECT IT TO THE NEXT RED DOT ON"
910  PRINT "ITS RIGHT.  HE HITS 'D' TO CONNECT IT"
920  PRINT "TO THE RED DOT BELOW IT. THE SECOND"
930  PRINT "PLAYER (BLUE) THEN DOES LIKEWISE."
940  PRINT
950  PRINT "RED'S AIM IS TO JOIN UP A VERTICAL"
960  PRINT "LINE DOWN THE BOARD."
970  PRINT "BLUE'S AIM IS TO JOIN UP A HORIZONTAL"
980  PRINT "LINE FROM LEFT TO RIGHT ACROSS THE BOARD"
```

待續…

```
990  PRINT
1000  PRINT "NO LINE MAY CROSS ANOTHER "
1010  PRINT
1020  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE  ";: NORMAL
1030  GET H$
1040  RETURN
3020 PITCH = 255:RISE = 1.0595 ^ 3: GOTO 3035
3030 PITCH = 120:RISE = 1 / (1.0595 ^ 3)
3035 D = 50
3040  FOR J = 1 TO 4
3050 PITCH = PITCH / RISE
3060  POKE 768,PITCH: POKE 769,D: CALL 770
3070  NEXT J
3080  RETURN
3090  REM  INITIAL POKES-----
3100  POKE 770,173: POKE 771,48: POKE 772,192: POKE 773,136: POKE 774,208
     : POKE 775,5: POKE 776,206: POKE 777,1: POKE 778,3: POKE 779,240: POKE
     780,9: POKE 781,202
3110  POKE 782,208: POKE 783,245: POKE 784,174: POKE 785,0: POKE 786,3: POKE
     787,76: POKE 788,2: POKE 789,3: POKE 790,96: RETURN
```

股票市場(Stock Market)



兩種股票行情範例

這是一個介於兩位擁有雄厚財力的企業大亨之間的高解析度智慧戰—那就是您與您的朋友。假設您們兩位都炒股票，而且都很注意股票市場股價指數的上漲或下跌（假定股價是隨機變動的），您們每個人都要買三次賣三次，最後賺得最多的人是贏家。記住，根據一位財經專家的分析，若想致富您必須具備三項條件：(1)沒受過教育，(2)沒有頭腦，(3)低價買進，高價賣出。

兩位遊戲者肩併肩坐在鍵盤前面，左邊的遊戲者按" E "賣股票，按" X "

買股票，而右邊的遊戲者按“ I ”賣股票，按“ M ”買股票。要記住這幾個鍵並不難：按下面一排鍵（ X或M ）表示要低價買進，按上面一排鍵（ E或I ）表示要高價賣出。

主迴圈由170那一列開始，按股票隨機上漲和下跌在螢光幕上描繪出股價情形。240那一列檢查是否有鍵被按下，而290至340那幾列則檢查到底那個鍵被按一是否左邊遊戲者按了“ X ”要買或“ E ”要賣，抑或右邊遊戲者按“ M ”要買或“ I ”要賣。350至420那幾列保證每位遊戲者一定是先買再賣，這是用來避免遊戲者抓住市場低迷時機一口氣買完，再挑高價時連續賣光。這種強迫先買再賣的作法迫使遊戲者眞正在玩股票，而不是孤注一擲式的投機，也可以防止他們買空賣空。每一次交易都會利用1200至1370那幾列的描字副程式在該項交易之後標下“ BUY ”（ 買 ）或“ SELL ”（ 賣 ）。

遊戲進行的狀況，包括每位遊戲者的盈餘，是在520至690那幾列中PRINT在螢光幕底部的。贏家當然是有較多盈餘的那個人。

本遊戲是爲投機者而寫的，不是爲那些喜歡穩穩汲取蠅頭小利及快速回收的人而寫的；也不是爲那些小心謹慎，只喜歡分享別人賭博的樂趣，自己從來不冒險的人而寫的。不過，在現實世界裡，股票市場最富有的人是那些股票經紀人，他們並不是靠賭股票價格的波動致富的，而是腳踏實地從投資者或那些自認爲可預測股價動向的投資者的佣金累積下來的。有一個古老的鄉下土包子故事說，有一次該鄉下人去遊覽華爾街，當遊到最後一站一東河時，導遊說“看！那些都是股票經紀商的遊艇”這位土包子問道，“他們客戶的遊艇在那裡？”。

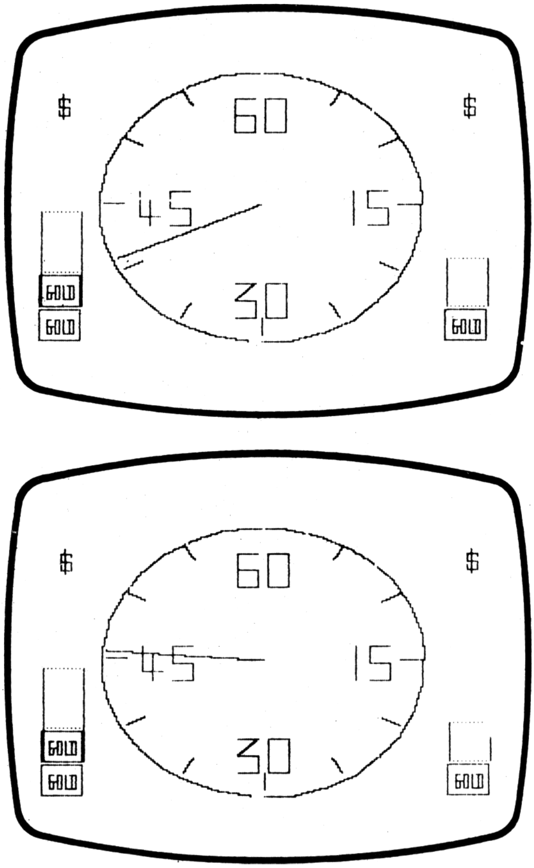
您在把錢投資到實際股票市場前可以先玩幾次這個股票市場遊戲，它可供您練習如何很快做決定，以及協助您判斷投資到底是一項策略遊戲，抑或純粹靠運氣。

Stock Market

```
10  REM STOCK MARKET-----
20  TEXT : HOME : VTAB 10
30  PRINT "WANT INSTRUCTIONS (Y/N) ?  "
40  GET Y$: IF Y$ = "Y" THEN  GOSUB 800
50  GOSUB 520: REM  SUBTITLES
55 ZL(1) = 1:ZL(2) = 1: REM MUST BUY BEFORE SELLING
60 W$ = "EXIM": REM BUY & SELL KEYS-----
70 Y = 80: REM  STARTING PRICE
80 W = 2: REM  WIDTH OF LETTERS IN BUY/SELL
90 F = 0
100  REM PLOT FRAME-----
110  HGR : HCOLOR= 7
120  HPLOT 0,0 TO 245,0 TO 245,159 TO 0,159 TO 0,0
130  FOR V = 0 TO 160 STEP 20
140  HPLOT 0,V TO 10,V
150  HPLOT 235,V TO 245,V
160  NEXT V
170  REM  MAIN LOOP-----
180  FOR X = 4 TO 245
190  HPLOT X,Y
200 Y = Y + 10 *  RND (1) - 10 *  RND (1): REM PRICE VARIES RANDOMLY
210  IF Y > 156 THEN Y = Y - 20 *  RND (1)
220  IF Y < 3 THEN Y = Y + 20 *  RND (1)
230  HPLOT  TO X,Y
240 KEY =  PEEK ( - 16384)
250  IF KEY > 127 THEN YY =  INT ((159 - Y) / 2): GOSUB 290
260  IF N(1) < 6 OR N(2) < 6 THEN  NEXT X: REM ELSE GAME OVER
270  IF N(1) < 6 OR N(2) < 6 THEN  GOTO 110
280  GOTO 990: REM  END OF GAME
290  REM  CHECK WHICH KEY PRESSED:E,X,I,M?-----
300  POKE  - 16368,0: REM  CLEAR KEYBOARD
310 KEY = KEY - 128
320  FOR J = 1 TO 4
330  IF KEY =  ASC ( MID$ (W$,J,1)) THEN CHOICE = J:P = (CHOICE > 2.5) +
     1
340  NEXT J
350  REM FORCE BUY & SELL TO ALTERNATE---
360 HIT(P) = CHOICE - 2 * (CHOICE > 2.5): REM 1=BUY, 2=SELL
370  IF HIT(P) = ZL(P) THEN  FOR M = 1 TO 10:BUZZ =  PEEK ( - 16336) -  PEEK
     ( - 16336): NEXT M: RETURN
380 ZL(P) = HIT(P): REM   ZL=PREVIOUS BUY/SELL KEYPRESS
390  GOSUB 430
400  IF F = 0 THEN  GOSUB 700
410 F = 0
420  RETURN
430  REM  RESPOND TO KEY PRESS -----
440 BUYSELL(CHOICE) = BUYSELL(CHOICE) + 1
450  IF BUYSELL(CHOICE) > 3 THEN BUYSELL(CHOICE) = 3: CALL  - 1059: CALL
      - 1059:F = 1: RETURN
460 SIGN = 1: IF CHOICE = 2 OR CHOICE = 4 THEN SIGN =  - 1
470 PROFIT(P) =  INT (PROFIT(P) + SIGN * YY)
480  VTAB 20 + CHOICE: HTAB 3 * BUYSELL(CHOICE) + 12
490  PRINT YY;
500  VTAB 19 + 2 * P: HTAB 26: PRINT "PROFIT=";PROFIT(P);"  "
510  RETURN
520  REM  SUBTITLES -----
530  TEXT : HOME
540  VTAB 21: PRINT "#1:E=SELL AT";
550  GOSUB 640
560  VTAB 22: PRINT "  :X= BUY AT";
570  GOSUB 640
580  VTAB 23: PRINT "#2:I=SELL AT";
590  GOSUB 640
600  VTAB 24: PRINT "  :M= BUY AT";
610 F = 1
620  GOSUB 640
630  RETURN
640  FOR G = 1 TO 3
650  HTAB 3 * G + 12
660  PRINT ".";
670  NEXT G
680  IF F < 1 THEN  PRINT
```

```
690  RETURN
700  REM PRINT 'BUY' OR 'SELL'-----
710 TX = X:TY = Y
720  HCOLOR= 7
730 N(P) = N(P) + 1
740 Y = 90 * (P = 2) + 4 + 8 * N(P)
750  HPLOT TX,TY TO TX,Y
760  IF CHOICE = 1 OR CHOICE = 3 THEN  GOSUB 1200: GOSUB 1250: GOTO 780: REM
     SELL
770  GOSUB 1290: REM BUY
780 X = TX:Y = TY
790  RETURN
800  REM  INSTRUCTIONS -----
810  HOME : VTAB 5
820  PRINT "  *****       STOCK MARKET      *****"
830  PRINT : PRINT
840  PRINT "THIS IS A GAME OF HIGH FINANCE."
850  PRINT
860  PRINT "STOCK MARKET PRICES ARE PLOTTED ON THE"
870  PRINT "SCREEN. EACH PLAYER MUST MAKE 3 BUYS &"
880  PRINT "3 SELLS, IN ALTERNATION (BUZZ SOUNDS IF"
890  PRINT "YOU DON'T ALTERNATE)."
900  PRINT : PRINT "TO WIN: BUY CHEAP, SELL DEAR."
910  PRINT : PRINT " LEFT PLAYER HITS E TO SELL, X TO BUY."
920  PRINT : PRINT "RIGHT PLAYER HITS I TO SELL, M TO BUY."
930  PRINT
940  INVERSE
950  PRINT "HIT ANY KEY TO CONTINUE."
960  NORMAL
970  GET HIT$
980  RETURN
990  REM  END OF GAME -----
1000  VTAB 24: HTAB 28
1010  FLASH
1020  PRINT " GAME OVER ";
1030  NORMAL
1040  CALL  - 1059: CALL  - 1059: CALL  - 1059
1050  FOR T = 1 TO 3000: NEXT T
1060  HOME : VTAB 21
1070  FOR P = 1 TO 2
1080  PRINT "PLAYER # ";P;" MADE A NET PROFIT OF ";PROFIT(P)
1090  NEXT P
1100 K = (PROFIT(2) > PROFIT(1)) + 1
1110  PRINT "PLAYER # ";K;" HAS WON !!"
1120  PRINT "WANT ANOTHER GAME (Y/N) ";
1130  GET Y$
1140  IF Y$ = "Y" THEN  FOR J = 1 TO 4:BUYSELL(J) = 0:PROFIT(J) = 0:N(J) =
     0: NEXT J:MOVES = 0: GOTO 50
1150  HOME : VTAB 21: PRINT "HIT 'ESC' TO EXIT."
1160  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU."
1170  GET G$
1180  IF G$ =  CHR$ (27) THEN  TEXT : HOME : VTAB 5: PRINT "BYE. THANKS F
     OR PLAYING.": END
1190  PRINT : PRINT  CHR$ (4);"RUN HELLO": END : REM   RETURN TO MENU
1200  REM  SELL-----
1210  HPLOT X + W,Y - W TO X - W,Y - W TO X - W,Y TO X + W,Y TO X + W,Y +
     W TO X - W,Y + W
1220 X = X + 8
1230  HPLOT X + W,Y - W TO X - W,Y - W TO X - W,Y + W TO X + W,Y + W
1240  HPLOT X - W,Y TO X + W,Y
1250  REM  L---
1260 X = X + 8
1270  HPLOT X - W,Y - W TO X - W,Y + W TO X + W,Y + W
1280  RETURN
1290  REM  BUY-----
1300  HPLOT X - W,Y - W TO X - W,Y + W TO X + W,Y + W TO X + W,Y TO X + W
     / 2,Y TO X + W / 2,Y - W TO X - W,Y - W
1310  HPLOT X - W,Y TO X + W,Y
1320 X = X + 8
1330  HPLOT X - W,Y - W TO X - W,Y + W TO X + W,Y + W TO X + W,Y - W
1340 X = X + 8
1350  HPLOT X - W,Y - W TO X,Y TO X,Y + W
1360  HPLOT X + W,Y - W TO X,Y
1370  RETURN
```

黃金大拍賣(Gold Auction)



黃金大拍賣所用的計時鐘

這是一個供兩人玩的財務策略遊戲，每位遊戲者要由 $.120,000 元起家，設法在大拍賣中靠買金條來增加資產。每一場拍賣會提出 8 條金條來拍賣，每一條在遊戲結束時以 $ 60,000 元贖回。因爲對每一條金條您所能出的最高價爲 $ 60,000 元，因此每一條都是很合算的，所以您應該儘可能地買，但很不幸地，您並沒有那麼多資本可買，尤其是您的對手還一直在出價逼您。

這個拍賣會不是一般傳統的拍賣會，由出價者漸漸把價錢抬到所有競爭者退

出爲止；它是一種"荷蘭式拍賣法"，由高價開始漸漸降低，直到有人出價爲止，第一個出價的人得到貨物！螢光幕上有一面指針可由60倒數至0的鐘，指針所指的位置就是金條的價格；亦即由6萬開始漸減至零，最先按控制器旋鈕的人將以指針所示的價格得到金條。如果您按得太遲，您的對手可能比您先按而得到金條；不過，如果您按得太早，您可能會以高價買到金條，那就不太好，因爲您可能很快就會用完那些錢，讓您的對手以極低價格買到第6，7和第8條金條，因爲您已經沒有足夠金錢與他競價了，最佳的策略是剛剛好在您的對手之前按下按鈕；當然，最好是手持著控制器避開他的視線。這整個遊戲就是一種"戰爭心理"的遊戲。

本遊戲是以高解析度圖形執行的，您按空白鍵就可開始一場拍賣會。描繪可嘀嗒響的計時鐘副程式在第一章中的"圓，螺旋線和鐘面"一節中就說明過了。當金條被買走時它會被畫在螢光幕上買方那一邊，他所剩餘的資本也會被畫在金條上面的方塊中；因此，看那個方塊的總高度就可看出各遊戲者的總資產（金條加現金）。

本遊戲是根據Toronto的Bob Swanson先生的構想改寫的。

Gold Auction

```
10  REM  *********************
20  REM       GOLD AUCTION
30  REM     BY STUART ANSTIS
40  REM  *********************
50 GAME = 0
60 JMAX = 8: REM  NUMBER OF BARS  AUCTIONED
70  DIM CAPITAL(2),ASSETS(2),BARS(2),FLAG(2)
80  FOR P = 1 TO 2
90 ASSETS(P) = 0:BARS(P) = 0:CAPITAL(P) = 120000
100 FLAG(P) = 0
110  NEXT P
120  IF GAME > 0 THEN  GOTO 160
```

```
125  GOSUB 2000: REM  PLAYERS' NAMES
130  PRINT "WANT INSTRUCTIONS (Y/N) ?"
140  GET I$
150  IF I$ = "Y" THEN  GOSUB 960
160  HOME : VTAB 21
170 PI = 3.14159:R = 80: HCOLOR= 3
180  HGR : REM   DRAW A CIRCLE -----
190  FOR A = 0 TO PI * 2 STEP 0.2
200  IF A = 0 THEN  HPLOT 140 + R *  SIN (A),R + R *  COS (A): GOTO 220
210  HPLOT  TO 140 + R *  SIN (A),R + R *  COS (A)
220  NEXT A
222 R = R - 2
224  GOSUB 230
226  GOTO 460
230  REM   DRAW CLOCK MARKERS -----
240  FOR C = 0 TO PI * 2 STEP PI / 6
250  HPLOT 140 + R *  SIN (C),R + R *  COS (C) TO 140 + (R - 10) *  SIN (
     C),R + (R - 10) *  COS (C)
260  NEXT C
270  REM  PLOT CLOCK DIGITS -----
280 H = 10:V = 10
290 X = 137:Y = 15
300  GOSUB 1750: REM  6
310 X = X + 15
320  GOSUB 1590: REM  0
330 X = 90:Y = 70
340  GOSUB 1680: REM  4
350 X = X + 15
360  GOSUB 1720: REM  5
370 X = 137:Y = 125
380  GOSUB 1650: REM  3
390 X = X + 15
400  GOSUB 1590: REM  0
410 X = 185:Y = 70
420  GOSUB 1620: REM  1
430 X = X + 17
440  GOSUB 1720: REM  5
445  RETURN
450  REM    START CLOCK ----------
460  FOR J = 1 TO JMAX
470  FOR B = 0 TO 2 * PI STEP .05
480 SOUND =  PEEK ( - 16336) -  PEEK ( - 16336)
490  FOR I = 3 TO 0 STEP  - 3
500  HCOLOR= I
510  IF I = 0 THEN  GOTO 560: REM  TEST BUTTONS
520  HPLOT 140,R TO 140 - R *  SIN (B),R - R *  COS (B)
530  NEXT I,B
540  PRINT "  ***  NO SALE !!!   ***"
550  GOTO 720
560  REM  TEST BUTTONS ----------
570  FOR P = 1 TO 2
580 X =  PEEK ( - 16288 + P)
590  IF X > 127 THEN  GOTO 620
600  NEXT P
610  GOTO 520
620  REM  PRINTOUT ----------
630 PRICE =  INT ((2 * PI - B) * (6 / 2 * PI) + 0.5) * 1012
640  IF CAPITAL(P) < PRICE THEN  GOTO 1330: REM  CAN'T AFFORD IT
650  PRINT NAME$(P);" JUST PAID ";PRICE;" DOLLARS"
660 CAPITAL(P) = CAPITAL(P) - PRICE
670 BARS(P) = BARS(P) + 1
680  PRINT "FUNDS LEFT =",CAPITAL(1),CAPITAL(2)
690 X = 0
700  GOSUB 1110: REM  DRAW GOLD BARS
710  IF J = JMAX THEN  GOTO 760
720  INVERSE : PRINT "  HIT SPACE BAR FOR AUCTION #  ";J + 1;"     ": NORMAL

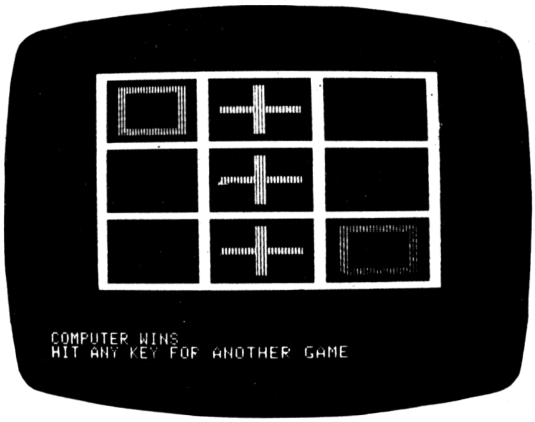
730  GET N$
740  HCOLOR= 0: HPLOT 140,R TO 140 - R *  SIN (B),R - R *  COS (B)
750  HCOLOR= 3: GOSUB 230: REM  RE-PLOT CLOCK MARKERS AND DIGITS
760  NEXT J
770  REM  FINAL PRINTOUT ----------
780  TEXT : HOME : VTAB 2: HTAB 10
782  PRINT "FINAL BALANCE SHEET"
784  PRINT
786  FOR M = 1 TO 8: PRINT "*****";: NEXT M
788  HTAB 15: PRINT "$ (X 1000)"
790  FOR P = 1 TO 2
792  PRINT NAME$(P): PRINT
794  HTAB 5: PRINT BARS(P);" GOLD BARS AT $60K...  ",60 * BARS(P)
796  HTAB 5: PRINT "CASH....   ",CAPITAL(P) / 1000
798 ASSETS(P) = 60 * BARS(P) + CAPITAL(P) / 1000
```

待續

```
800  HTAB 5: PRINT "TOTAL ASSETS.....      ",ASSETS(P)
802  PRINT
804  NEXT P
860  PRINT : PRINT
870  IF ASSETS(1) > ASSETS(2) THEN  PRINT NAME$(1);" HAS WON !!! ": GOTO
     910
880  IF ASSETS(1) = ASSETS(2) THEN  PRINT "THE GAME WAS A DRAW !!!": GOTO
     910
890  PRINT NAME$(2);" HAS WON !!!"
900  PRINT
910  PRINT "WANT ANOTHER GAME? (Y/N)"
920 GAME = GAME + 1
930  GET Y$
932  IF Y$ = "Y" THEN  GOTO 80
934  TEXT : HOME : VTAB 10
936  PRINT "HIT 'ESC' TO EXIT"
938  PRINT : PRINT "OR, HIT ANY OTHER KEY TO RETURN TO MENU"
940  GET H$
942  IF H$ =  CHR$ (27) THEN  HOME : VTAB 10: PRINT "OK. THANKS FOR PLAYI
     NG.": END
944  PRINT : PRINT  CHR$ (4);"RUN HELLO"
946  END
960  REM  INSTRUCTIONS ----------
970  PRINT : PRINT : PRINT : PRINT
980  PRINT "THIS IS AN AUCTION OF GOLD BARS."
990  PRINT JMAX;" GOLD BARS WILL BE AUCTIONED.  EACH BAR IS WORTH 60 THOU
     SAND DOLLARS."
1000  PRINT "THE PRICE IS SHOWN BY THE CLOCK HAND"
1005  PRINT "WHICH TICKS DOWN FROM 60 THOUSAND TO": PRINT "ZERO DOLLARS.
     ";
1010  PRINT "THE FIRST PLAYER TO PUSH HIS BUTTON BUYS THE GOLD BAR.  IF Y
     OU"
1015  PRINT "PUSH TOO SOON YOU WILL PAY A HIGH PRICE.  ";
1020  PRINT "IF YOU PUSH TOO LATE THE OTHER PLAYER": PRINT "GETS THE GOLD
     ";
1030  PRINT "TRY TO PUSH YOUR BUTTON"
1035  PRINT "JUST BEFORE THE OTHER PLAYER.  "
1040  PRINT "IF YOU OVERPSPEND YOU GO BANKRUPT AND": PRINT "LOSE THE GAME
     "
1050  PRINT "EACH PLAYER STARTS OFF WITH ";CAPITAL(1) / 1000;" THOUSAND D
     OLLARS. "
1060  PRINT : PRINT : PRINT "     $$$   GOOD LUCK !!!    $$$"
1070  PRINT
1080  INVERSE : PRINT "  **  HIT SPACE BAR FOR AUCTION # ";J + 1;"  ** ":
      NORMAL
1090  GET N$
1100  RETURN
1110  REM  DRAW GOLD BARS ----------
1120  FOR Q = 1 TO 0 STEP  - 1
1130 YY = 160 - 20 * (BARS(P) - Q) - (CAPITAL(P) + Q * PRICE) / 3000
1150 Y = 160 - 20 * BARS(P)
1160 XX = 130 + 100 * (2 * P - 3)
1162 H = 5:V = 5
1165 AA = XX + 15:BB = 15
1166  HCOLOR= 7
1168  IF Q = 1 THEN  GOSUB 1920: REM  $
1169 HC = 5 - 5 * Q: HCOLOR= HC
1170  HPLOT XX + 1,Y - 2 TO XX + 1,YY TO XX + 21,YY TO XX + 21,Y - 2 TO X
     X + 1,Y - 2
1180  NEXT Q
1190  FOR P = 1 TO 2
1200  HCOLOR= 7
1210  HPLOT XX,Y TO XX,Y + 18 TO XX + 20,Y + 18 TO XX + 20,Y TO XX,Y
1220  NEXT P
1230 X = XX + 6:Y = Y + 6
1240 H = 2:V = 4
1250  GOSUB 1840: REM  G
1260 X = X + 4
1270  GOSUB 1590: REM  O
1280 X = X + 4
1290  GOSUB 1810: REM  L
1300 X = X + 3
1310  GOSUB 1880: REM  D
1320  RETURN
1330  REM   CAN'T AFFORD IT -----
1340  GOSUB 1400: REM  SOUND EFFECTS
1350  PRINT "OUT OF FUNDS -- FINED $10000"
1360 CAPITAL(P) = CAPITAL(P) - 10000
1370  IF FLAG(P) > 0 THEN  GOTO 1460
1380 FLAG(P) = FLAG(P) + 1
1390  GOTO 680
1400  REM  SOUND EFFECTS ------
1410 S =  - 16336
1420  FOR M = 1 TO 25
```

```
1430 SOUND =  PEEK (S) -  PEEK (S) -  PEEK (S) -  PEEK (S)
1440  NEXT M
1450  RETURN
1460  REM  FINAL BANKRUPTCY PRINTOUT -----
1470  TEXT
1480  FOR L = 1 TO 50: PRINT : NEXT L
1490  PRINT NAME$(P): PRINT " HAS TWICE OVER-BID WITH INSUFFICIENT": PRINT
 "FUNDS."
1500  PRINT "HE IS BANKRUPT."
1510  PRINT : PRINT "THEREFORE,"
1520  PRINT : PRINT NAME$(3 - P): PRINT "WHILE NO ANGEL HIMSELF, IS DECLA
 RED...."
1530  PRINT : PRINT "THE WINNER !!!"
1540  PRINT : PRINT : PRINT
1550 GAME = GAME + 1
1560  GOTO 910
1580  REM  PLOT DIGITS -----
1590  REM  0 -----
1600  HPLOT X,Y TO X - H,Y TO X - H,Y + 2 * V TO X,Y + 2 * V TO X,Y
1610  RETURN
1620  REM  1 -----
1630  HPLOT X,Y TO X,Y + 2 * V
1640  RETURN
1650  REM  3 -----
1660  HPLOT X - H,Y TO X,Y TO X - H,Y + V TO X,Y + V TO X,Y + 2 * V TO X -
 H,Y + 2 * V
1670  RETURN
1680  REM  4 -----
1690  HPLOT X - H,Y TO X - H,Y + 1.5 * V TO X,Y + 1.5 * V
1700  HPLOT X - 0.5 * H,Y + 0.5 * V TO X - 0.5 * H,Y + 2 * V
1710  RETURN
1720  REM  5 -----
1730  HPLOT X,Y TO X - H,Y TO X - H,Y + V TO X,Y + V TO X,Y + 2 * V TO X -
 H,Y + 2 * V
1740  RETURN
1750  REM  6 -----
1760  HPLOT X,Y TO X - H,Y TO X - H,Y + 2 * V TO X,Y + 2 * V TO X,Y + V TO
 X - H,Y + V
1770  RETURN
1780  REM  NEW 4 -----
1790  HPLOT X,Y + V TO X - 2 * H,Y + V TO X - H,Y TO X - H,Y + 2 * V
1800  RETURN
1810  REM    L -----
1820  HPLOT X - H,Y TO X - H,Y + 2 * V TO X,Y + 2 * V
1830  RETURN
1840  REM  G -----
1850  HPLOT X,Y TO X - H,Y TO X - H,Y + 2 * V TO X,Y + 2 * V TO X,Y + V
1860  HPLOT X + 0.7 * H,Y + V TO X - 0.4 * H,Y + V
1870  RETURN
1880  REM  D -----
1890  HPLOT X - 1.2 * H,Y TO X,Y TO X,Y + 2 * V TO X - 1.2 * H,Y + 2 * V
1900  HPLOT X - 0.7 * H,Y TO X - 0.7 * H,Y + 2 * V
1910  RETURN
1920  REM  $ -----
1930  HPLOT AA,BB + V / 2 TO AA,BB TO AA - H,BB TO AA - H,BB + V TO AA,BB
 + V TO AA,BB + 2 * V TO AA - H,BB + 2 * V TO AA - H,BB + 1.5 * V
1940  HPLOT AA - H / 2,BB - V / 2 TO AA - H / 2,BB + 2.5 * V
1950  RETURN
2000  REM  GET PLAYERS' NAMES-----
2002  TEXT : HOME : VTAB 2
2004  PRINT "THIS IS A GAME OF STRATEGY FOR TWO": PRINT "PLAYERS."
2006  PRINT
2020 PL$ = "FIRST SECOND"
2030  FOR W = 1 TO 2
2040  PRINT  MID$ (PL$,1 + 6 * (W - 1),6);" PLAYER, PLEASE TYPE IN YOUR N
 AME & HIT RETURN"
2050  INPUT NAME$(W)
2060  PRINT : PRINT "THANK YOU": PRINT
2070  NEXT W
2080  PRINT "EACH PLAYER HAS A PADDLE. PLEASE TURN"
2090  PRINT "ONE OF THE PADDLE KNOBS NOW."
2100  FOR W = 0 TO 1
2110 PAD(W) =  PDL (W)
2120  IF  ABS (PAD(W) -  PDL (W)) > 1 THEN Q = W: GOTO 2150
2130  NEXT W
2140  GOTO 2100
2150  PRINT : PRINT "THANK YOU.  THAT IS PADDLE # ";Q;". GIVE IT TO ";NAM
 E$(W + 1)
2160  PRINT "(GIVE THE OTHER PADDLE TO ";NAME$(2 - W);")"
2170  PRINT : PRINT
2180  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE  ": NORMAL
2190  GET H$
2200  HOME : VTAB 10
2210  RETURN
```

井字棋(Tic-Tac-Toe)



這是一個描繪棋盤遊戲的入門遊戲

這是一種流傳於英國的古老，簡單又有名的" Noughts and Crosses "（圖與叉）遊戲，很適合用低解析度圖形來執行。

與電腦玩井字棋是很有趣的，它是一種很好又簡單的策略遊戲，變化也不多，很適合用來做人工智慧的研究。電腦的戰略詳註在程式的REM敍述中，您只要將程式LIST出來就可理解它是怎麼一回事。當遊戲者走了一步之後，電腦便會算一算，然後在極短的時間內（幾乎是瞬間的）做出反應。若想給遊戲者留點面子，可在程式中加入一段延時迴圈，使電腦看起來好像需要一些時間去思考：

```
FOR TIME = 1 TO 2000: NEXT TIME
```

下面是本遊戲的另外一種形式：假設您與電腦輪流挑選一個介於1與9（包含）之間的數目放在格子內，而且一旦某個數目被選過之後任何人都不能再選，此遊戲的目的是要選取三個數目使每一行，每一列與對角綫的和均爲15 。爲什麼說它是井字遊戲的另一種形式呢？請看看下面這個奇妙的數字組合：

```
8 1 6
3 5 7
4 9 2
```

每一列，每一行以及對角線的和都是 15，要挑選三個數目使其和爲 15 其實就是在棋盤上如何組成致勝的一排。您可以當作一個練習，寫個程式讓電腦利用這種方法來致勝。第一個數目挑選 5 是很好的戰術（想想看？）

愛丁堡大學人工智慧系（Department of Artificial Intelligence at the University of Edinburgh）的 Donald Michie 教授曾建造一部可逐漸學習玩井字棋的“火柴盒電腦”（Matchbox computer）。他利用好幾打火柴盒製作了一個很大的抽斗，每個火柴盒內放置一些標有一個 1 至 9 數目的彈珠。遊戲的進行包含盲目地從適當的火柴盒中選擇一個彈珠，而彈珠上的數目即指定下一步棋要走的方格。

井字棋僅有有限個可能的“狀態”；一個“狀態”代表遊戲期間某一時刻板上可能的樣子。下面是三個例子：

```
1.       2.       3.

O - X    - O X    X - X
- O -    O O -    - O -
- - X    - X -    O - -
```

在狀態 2 中，O 走了三步，而 X 走了兩步。在狀態 1 和 3 中，O 和 X 都走了兩步。事實上，狀態 3 只是將狀態 1 轉個直角，因此狀態 1 和 3 要算相同的狀態。（所有反射與旋轉能得出相同結果的都應看成相同狀態而予以忽略。）Michie 教授將遊戲中每一種可能的狀態分別畫在一張背了膠的標籤上，然後分別黏貼在一個火柴盒上，而每個火柴盒只包含那些代表各個可能方格（尚未填入）的彈珠。當遊戲進行到各個不同狀態時他就從標着該狀態的火柴盒中取出一粒代表電腦下一步要走的彈珠。他讓抽斗一直開着，直到遊戲結束。當遊戲結束時，如果電腦贏了那一局，他就額外將一粒標著他剛走那一步的彈珠放到各個開着的火柴盒中，使下一局走那一步棋的機會增加；如果電腦輸了，他就從各個開着的火柴盒

中取出一粒代表剛走那一步棋的彈珠，使下一次走那一步棋的機會減少。換句話說，如果贏了他就讓所有導致勝利的那幾步棋在將來更容易出現；如果失敗了，他就以相反的程序加以處罰。他發現剛開始時該火柴盒"機器"是隨便玩，但大約經過一百局以後它就很有成績了。

您可以試試看寫個程式來模擬Michie教授的火柴盒。

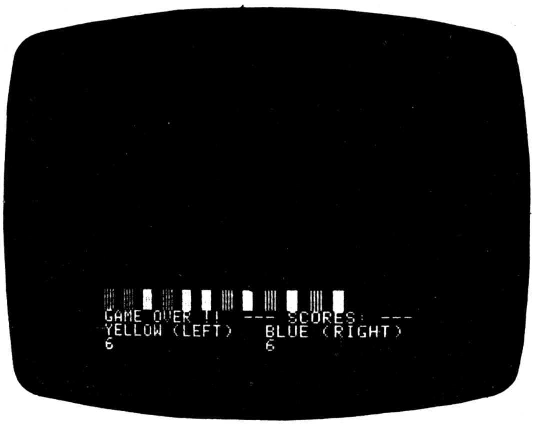
Tic-Tac-Toe

```
10  REM  TIC TAC TOE -----
20 H = 10:S = 3
25  REM  PLOT GRID -----
30  GR
40  COLOR= 15
50  FOR I = 0.5 * H TO 3.5 * H STEP H
60  HLIN 0.5 * H,3.5 * H AT I
70  VLIN 0.5 * H,3.5 * H AT I
80  NEXT I
90  REM : FN M(X) LEAVES 1 THROUGH 8 UNALTERED, AND CHANGES 9 TO 1 -----
100  DEF  FN M(X) = X - 8 *  INT ((X - 1) / 8)
110  VTAB 21
130  PRINT
140  GOSUB 850: REM  PRINT DIGITS
150  REM  MAIN LOOP -----
160 CM = 9
170  GOSUB 480: REM  COLLECT MOVES
180 CM =  FN M(YM + 1): REM  COMPUTER PLOTS TO RIGHT OF PLAYER'S LAST MOV
    E ---
190 P = CM
200  GOSUB 480
210  IF YM =  FN M(CM + 4) THEN 250
220 CM =  FN M(CM + 4): REM  COMPUTER PLOTS OPPOSITE ITS LAST MOVE ---
230  GOSUB 550
240  GOTO 580: REM  COMPUTER WINS ---
250 CM =  FN M(CM + 2): REM  COMPUTER PLOTS 2 SQUARES CLOCKWISE FROM LAST
     MOVE, I.E. NEXT CORNER OR NEXT MIDSQARE ---
260  GOSUB 480
270  IF YM =  FN M(CM + 4) THEN 310
280 CM =  FN M(CM + 4): REM   COMPUTER PLOTS OPPOSITE ITS LAST MOVE ---
290  GOSUB 550
300  GOTO 580
310  IF P / 2 =  INT (P / 2) THEN 350
320 CM =  FN M(CM + 7): REM  COMPUTER PLOTS NEXT SQUARE COUNTERCLOCKWISE
    FROM ITS LAST MOVE ---
330  GOSUB 550
340  GOTO 580
350 CM =  FN M(CM + 3)
360  GOSUB 480
370  IF YM =  FN M(CM + 4) THEN 410
380 CM =  FN M(CM + 4)
390  GOSUB 550
400  GOTO 580
410 CM =  FN M(CM + 6)
420  GOSUB 550
430  HOME : VTAB 21
440  PRINT "THE GAME IS A DRAW"
450  PRINT "HIT ANY KEY FOR ANOTHER GAME"
460  GET K$
470  GOTO 30
480  GOSUB 550
490  GOSUB 850
500  PRINT "YOUR MOVE ";
510  INPUT YM
520  ON YM GOSUB 640,650,660,670,680,690,700,710,720
530  GOSUB 780
540  RETURN
550  ON CM GOSUB 640,650,660,670,680,690,700,710,720
560  GOSUB 730
570  RETURN
580  CALL  - 1059: REM  BLEEP
590  HOME : VTAB 22: PRINT "COMPUTER WINS"
600  PRINT "HIT ANY KEY FOR ANOTHER GAME"
610  GET J$
620  GOTO 30
630  END
640 X = H:Y = H: RETURN
650 X = 2 * H:Y = H: RETURN
660 X = 3 * H:Y = H: RETURN
670 X = 3 * H:Y = 2 * H: RETURN
680 X = 3 * H:Y = 3 * H: RETURN
```

待續

```
690 X = 2 * H:Y = 3 * H: RETURN
700 X = H:Y = 3 * H: RETURN
710 X = H:Y = 2 * H: RETURN
720 X = 2 * H:Y = 2 * H: RETURN
730  REM  DRAW CROSS ---
740  COLOR= 9
750  HLIN X - S,X + S AT Y
760  VLIN Y - S,Y + S AT X
770  RETURN
780  REM  DRAW ZERO ---
790  COLOR= 2
800  HLIN X - S,X + S AT Y - S
810  HLIN X - S,X + S AT Y + S
820  VLIN Y - S,Y + S AT X - S
830  VLIN Y - S,Y + S AT X + S
840  RETURN
850  REM  PRINT DIGITS -----
860  HOME
870  VTAB 21
880  PRINT " 123","TIC TAC TOE"
890  PRINT " 894"
900  PRINT " 765"
910  RETURN
```

翻面(Turnover)



這是一種具挑戰性的邏輯選擇遊戲

這是一個供兩人玩的策略遊戲，有一列紅綠相間的小方塊排在螢光幕上。在遊戲開始之前，您要選擇小方塊的數目和遊戲中允許翻面的次數一這是很重要的。翻最後一次的人通常比較有利，因此翻面次數為偶數才公平，但奇數可用來讓第一位遊戲者佔點便宜。

每一步都包括一組小方塊的翻面，小方塊被翻面後會改變其顏色一紅變綠或綠變紅。0號遊戲者應該坐在左邊，設法獲得較多紅色小方塊，而 1號遊戲者應儘可能使小方塊都變成綠色。

假設現在有 12 塊小方塊， 0 號遊戲者應打入一個介於 1 與 12 之間的數目，比方說 5，這一步會翻轉第 1 至第 5 塊小方塊的顏色。1 號遊戲者打入一個數目，比方說 3，這一步會翻轉第 3 至第 12 塊小方塊的顏色（第 3，第 4 和第 5 塊被翻轉了兩次，因此它們又回復為原來的顏色；第 1 和第 2 塊以及第 6 至第 12 塊只翻

轉一次，因此它們的顏色與原來相反。）0號遊戲者所翻轉的小方塊在其選定號碼（包含）的左邊，而1號遊戲者所翻轉的小方塊在其選定號碼（包含）的右邊。

多奇妙的小遊戲！容易嗎？或者沒什麼意思？或者很難學會？

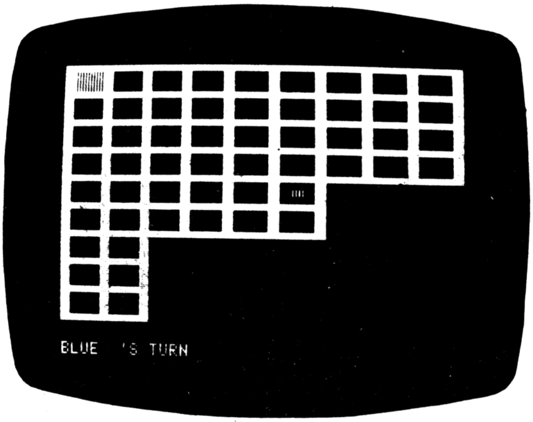
Turnover

```
10  REM   TURNOVER 6 IN APPLESOFT -----
20  TEXT : HOME : VTAB 10
30  INPUT "HOW MANY SQUARES (4--12)....";JMAX
40  IF JMAX > 12 THEN 30
50  INPUT "HOW MANY TURNS (ODD OR EVEN, 2--10)...";TMAX
60  IF TMAX < 2 OR TMAX > 10 THEN 50
70 P = 0: REM  PLAYER
80  HOME
90  GR
100  REM  PLOT INITIAL ROW OF COLORED SQUARES -----
110  FOR J = 1 TO JMAX
115 MOD = 2 * (J / 2 -  INT (J / 2))
120  COLOR= 2 + 11 * MOD
130  VLIN 37,39 AT 2 * J - 2
140  NEXT J
150  VTAB 22
160 R$ =  MID$ ("LEFT RIGHT",1 + 5 * P,5)
170  PRINT R$;" PLAYER, TURN OVER ALL SQUARES TO  ";R$;" (INCLUSIVE) OF W
     HAT NUMBER...";
180  INPUT K
190  IF K < 1 OR K > JMAX THEN 150
200  VTAB 21
210  FOR J = 1 TO JMAX
220  PRINT J;" ";: REM  WAS J MOD 10 ********
230  NEXT J
240  PRINT
250  REM  CHANGE COLORS ON EACH TURN -----
260  FOR I = K * P + 1 - P TO K - K * P + JMAX * P
270  COLOR= 15 -  SCRN( 2 * I - 2,39)
280  VLIN 37,39 AT 2 * I - 2
290  NEXT I
300 P = 1 - P: REM  CHANGE PLAYERS
310 TURN = TURN + 1
330  VTAB 21
340  HTAB 2 * JMAX - 2
```

待續

```
350  FOR M = 1 TO 29 - 2 * JMAX: PRINT " ";: NEXT M
360  PRINT "TURNS LEFT: ";TMAX - TURN
380  IF TURN = TMAX THEN  GOTO 400
390  GOTO 150
400  REM  END OF GAME -----
410  HOME
440  FOR S = 1 TO 16
450  IF  SCRN( 2 * S - 2,39) = 2 THEN SUM = SUM + 1
460  NEXT S
470  VTAB 22
475  PRINT "GAME OVER !!  --- SCORES: ---"
480  PRINT "YELLOW (LEFT)","BLUE (RIGHT)"
490  PRINT JMAX - SUM,SUM;
500  END
```

吞食蛋糕(Chomp)



最後吞食的人會拿到有毒的方塊

這個遊戲是加州大學柏克萊分校的David Gale發明的，它可算是Nim遊戲的二維空間版本。在Nim中，遊戲者輪流從幾堆火柴盒中拿出火柴盒，誰被迫拿到最後一盒火柴盒誰就是輸家。在本遊戲中，棋板是一塊被分成許多小方塊的大蛋糕；左上角是一塊有毒的方塊，誰拿到那一塊誰就是輸家。

若想大口吞食蛋糕可旋轉0號遊戲控制器選定一塊方塊，（旋轉控制器移動一個小指標沿著第一列從左至右，然後第二列，………，好像讀書一樣。）然後按下控制器按鈕，則在該方塊右邊（包含它自己）以及所有下面的方塊都會消失掉一這就是Chomp ！

您不得吞食那些已經被吞食過的方塊，如果您試圖這麼做，電腦會阻止您。

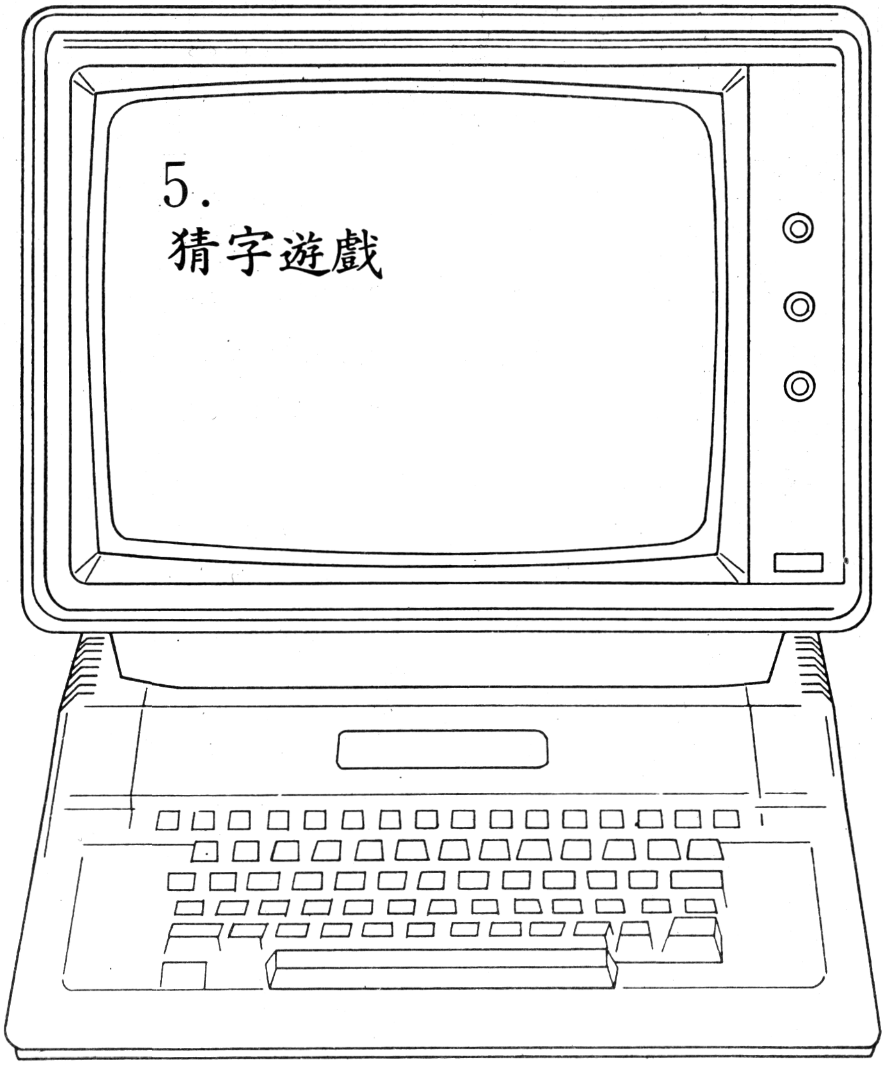
這個遊戲兩個人以上也可以玩，不過，兩個人比較好，因爲如果三，四個人一齊玩，雖然只有一個輸家，但不會有肯定的唯一贏家。

Chomp

```
10  REM ******************
20  REM *                *
30  REM *   APPLE CHOMP  *
40  REM *                *
50  REM *       BY       *
60  REM *                *
70  REM *  STUART ANSTIS *
80  REM *                *
90  REM ******************
100  REM
110  REM
120 P$ = "BLUE  YELLOW"
130  TEXT : HOME : VTAB 10
140  PRINT "WANT INSTRUCTIONS (Y/N) ?"
150  GET Y$
160  IF Y$ = "Y" THEN  GOSUB 980: REM  INSTRUCTIONS
170  TEXT : HOME : VTAB 10
180  PRINT "HOW BIG DO YOU WANT YOUR WAFFLE (3--9)  ";: GET S$
190 SIZE =  VAL (S$): IF SIZE < 3 OR SIZE > 9 THEN  PRINT "SORRY, MUST BE
     BETWEEN 3 AND 9": GOTO 180
200  REM  DRAW GRID-----
210  GR : COLOR= 15
220  FOR I = 0 TO 4 * SIZE STEP 4
230  HLIN 0,4 * SIZE AT I
240  VLIN 0,4 * SIZE AT I
250  NEXT I
260  COLOR= 9
270  HLIN 1,3 AT 1: HLIN 1,3 AT 2: HLIN 1,3 AT 3
280  REM  MAIN LOOP-----
290 P = 0: REM  PLAYER #
300 DIV = 256 / (SIZE * SIZE)
310 PADDLE =  PDL (P) / DIV
320  VTAB 21: HTAB 1: PRINT  MID$ (P$,1 + 6 * P,6);"'S TURN"
330 POLD =  PDL (P)
340 X =  INT (SIZE * (PADDLE / SIZE -  INT (PADDLE / SIZE))): REM  PADDLE
     MOD SIZE
350 Y =  INT (PADDLE / SIZE)
360 X = 4 * X + 2
```

待續…

```
370 Y = 4 * Y + 2
380  COLOR= 2 + 11 * P: REM  BLUE OR YELLOW
390  PLOT X,Y
400  IF  ABS (POLD -  PDL (P)) > 1 THEN  GOTO 440
410  IF  PEEK ( - 16287 + P) > 127 THEN  GOSUB 630
420  GOTO 400
430  REM  ERASE OLD POSITION -----
440  COLOR=  SCRN( X + 1,Y + 1)
450  PLOT X,Y
460  GOTO 310
470  REM  CHOMP -----
480 EDGE = 4 * SIZE + 2
490  COLOR= 0
500  FOR I = X + 2 TO EDGE STEP 4
510  FOR J = Y + 2 TO EDGE STEP 4
520  VLIN J - 3,EDGE AT I
530  HLIN I - 3,EDGE AT J
540  NEXT J,I
550  IF X = 2 THEN  VLIN Y - 1,EDGE AT 0
560  IF Y = 2 THEN  HLIN X - 1,EDGE AT 0
570  IF X = 2 AND Y = 2 THEN 830: REM  GAME OVER
580 P = 1 - P: REM  PLAYER #
590  GOSUB 760: REM  CHECK SIZE
600  COLOR= 9
610  IF  PEEK ( - 16268 + P) > 128 THEN  RETURN
620  GOTO 610
630  REM  CHECK IF MOVE IS LEGAL -----
640  FOR M = 1 TO 10:SOUND =  PEEK ( - 16336): NEXT M
650  HOME
660  IF  PEEK ( - 16287 + P) < 128 THEN 680
670  GOTO 660
680  IF  SCRN( X + 2,Y) = 15 THEN 740
690  CALL  - 1059
700  VTAB 22: PRINT "UNFAIR! YOU CAN'T CHOMP AIR!"
710  PRINT "MOVE YOUR PADDLE, "; MID$ (P$,1 + 6 * P,6);" -- TRY AGAIN"
720  IF  PEEK ( - 16286) < 128 THEN  RETURN
730  GOTO 720
740  GOSUB 470: REM  CHOMP
750  RETURN
760  REM  CHECK SIZE -----
770  FOR I = 39 TO 0 STEP  - 1
780  IF  SCRN( 0,I) = 15 OR  SCRN( I,0) = 15 THEN 800
790  NEXT I
800 SIZE = I / 4
810 DIV = 256 / (SIZE * SIZE)
820  RETURN
830  REM  GAME OVER -----
840  CALL  - 1059: CALL  - 1059
850  GR
860  HOME : VTAB 21
870  PRINT "  *****      GAME OVER  !!!      *****"
880  PRINT  MID$ (P$,1 + 6 * (1 - P),6);" HAS WON !!"
890  PRINT "WANT ANOTHER GAME (Y/N) ?"
900  GET Y$
910  IF Y$ = "Y" THEN  GOTO 170
920  TEXT : HOME : VTAB 10
930  PRINT "HIT 'ESC' TO EXIT"
940  PRINT : PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
950  GET H$
960  IF H$ =  CHR$ (27) THEN  HOME : VTAB 5: PRINT "OK. THANKS FOR PLAYIN
     G.": END
970  PRINT : PRINT  CHR$ (4);"RUN HELLO"
980  REM  INSTRUCTIONS -----
990  HOME
1000  PRINT : HTAB 15: PRINT "APPLE CHOMP"
1010  PRINT
1020  FOR M = 1 TO 8: PRINT "*****";: NEXT M
1030  PRINT
1040  PRINT "A GAME OF STRATEGY FOR TWO PLAYERS."
1050  PRINT
1060  PRINT "A GIANT WAFFLE WILL APPEAR ON THE SCREEN YOU CHOOSE ITS SIZE
       FROM 3 X 3 TO"
1070  PRINT "9 X 9 SQUARES.  THE TOP LEFT CORNER"
1080  PRINT "SQUARE IS POISON.  IF YOU BITE IT, YOU": PRINT "LOSE!!"
1090  PRINT : PRINT "EACH PLAYER TURNS HIS PADDLE TO SELECT ASQUARE.";
1100  PRINT " (THE LITTLE POINTER MOVES ALONG"
1110  PRINT "THE TOP ROW, THEN THE SECOND ROW, LIKE"
1120  PRINT "READING A BOOK).  PRESS THE BUTTON TO"
1130  PRINT "CHOMP OFF ALL SQUARES BELOW AND TO THE"
1140  PRINT "RIGHT.  THE PLAYER WHO HAS TO CHOMP THE"
1150  PRINT "LAST, POISONED SQUARE (TOP LEFT CORNER)"
1160  PRINT "IS THE LOSER."
1170  PRINT : PRINT
1180  INVERSE : PRINT " HIT ANY KEY TO CONTINUE " : NORMAL
1190  GET H$
1200  RETURN
```



劊子手(Hangman)

這個猜字遊戲是一個人人喜愛的古老遊戲，此處是用高解析度圖形來做，其玩法爲先由一位遊戲者偸偸打入一個字，然後讓第二位遊戲者來猜。（如果您想趁機練習DOS的檔案處理技巧，可將程式加以修改使它從磁碟片上READ（讀取

）一個稱爲WORDLIST 1的文字檔，並從中隨機挑選一個字。）

第二位遊戲者要設法一次打入一個字母來猜出那個字，如果他打入的字母包含在那個字中，程式會依照它在該字內的位置顯示在螢光幕上。如果打入的字母不在該字內，則帶有一個絞刑犯的絞首台就會開始出現，每猜錯一次便會有一部份肢體加上去，當錯了七次之後，該遊戲者就“死”了。

爲了幫助遊戲者記憶，所有字母都會印在螢光幕底部，只要遊戲者每猜一次，該字母便會從字母表中消失，這樣可避免遊戲者重複猜同一個字母。

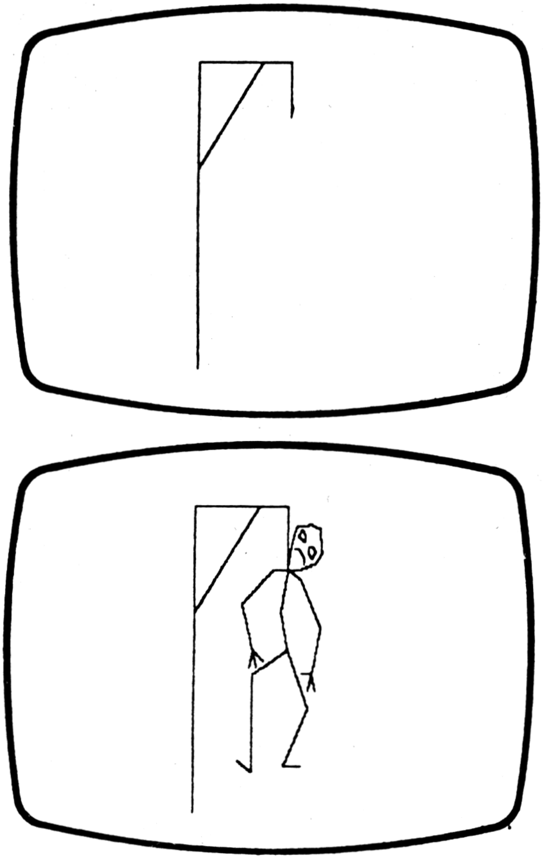
絞首台繪圖部份是本遊戲的重點，不過，您若想偸懶省下一些打字和偵錯工作，可省略繪圖部份：只要不打入570至870那幾列，並將550那一列從

```
550 ON FAIL GOSUB 570, 610, 670, 710, 750,
780, 810
```

改爲：

```
550 HPLOT 39 * FAIL, 0 TO 39 * FAIL, 159
```

當您每猜錯一次，您將會看到一條接一條地出現垂直線，不再是絞首台；當您看到七條線時您就“死”了。



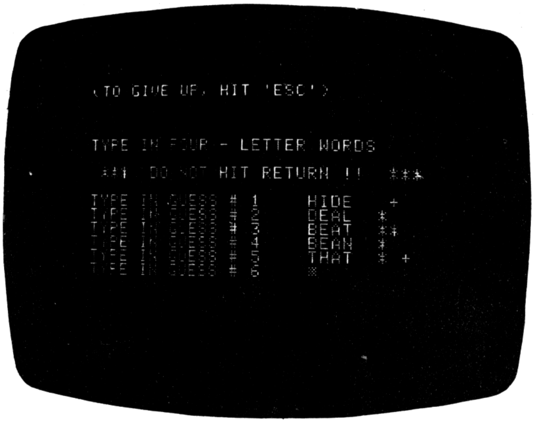
圖形剛開始時沒什麼，但很快地就會變得很恐怖

Hangman

```
10   REM     HANGMAN 15 -----
20   DIM W$(20),GUESS$(26),KNOWN$(20),BLANK(26)
30   REM  FIRST PLAYER TYPES TARGET WORD -----
35   TEXT : HOME : VTAB 10
40   PRINT "THINK OF A WORD FOR THE OTHER PLAYER TO GUESS."
60 FAIL = 0
80   PRINT
90   PRINT "PLEASE TYPE IN YOUR WORD AND HIT RETURN."
100   INPUT M$
110 N =  LEN (M$)
120   FOR J = 1 TO N:KNOWN$(J) = "=":W$(J) =  MID$ (M$,J,1): NEXT J
140   PRINT : PRINT
150   HGR
160   HCOLOR= 7
170   REM  SECOND PLAYER GUESSES A LETTER -----
180   FOR G = 1 TO 26
190   VTAB 23: PRINT "MAKE GUESS #";G
200   GET GUESS$(G)
210   VTAB 21: HTAB 15
220   FOR J = 65 TO 90: PRINT  CHR$ (J);: NEXT J
230   FOR K = 1 TO G
240 BLANK(K) =  ASC (GUESS$(K)) - 50
250   VTAB 21: HTAB BLANK(K)
260   PRINT " "
270   NEXT K
```

```
280  REM   CHECK IF GUESS CORRECT -----
290  FOR J = 1 TO N
300  IF GUESS$(G) = W$(J) THEN KNOWN$(J) = W$(J):FLAG = 1
310  VTAB 22: HTAB J: PRINT KNOWN$(J)
320  NEXT J
330  REM  IF ERROR THEN DRAW LIMB -----
340  IF FLAG = 0 THEN  GOSUB 530
350 FLAG = 0
360  IF FAIL = 7 THEN  GOTO 420
370  FOR J = 1 TO N
380  IF KNOWN$(J) = "=" THEN  GOTO 410
390  NEXT J
400  GOTO 510
410  NEXT G
420  PRINT : PRINT
425  CALL  - 1059
430  FLASH
440  PRINT "    ***   YOU ARE DEAD   !!!    ***"
450  NORMAL
460  PRINT
470  PRINT "WANT ANOTHER GAME (Y/N) ?"
480  GET Y$
490  IF Y$ = "Y" THEN  GOTO 30
492  PRINT "WANT TO RETURN TO MENU (Y/N) ?"
494  GET K$
496  IF K$ = "Y" THEN  PRINT : PRINT  CHR$ (4);"RUN HELLO"
498  TEXT : HOME : VTAB 10
499  PRINT "OK. THANKS FOR PLAYING."
500  END
510  PRINT "YOU GUESSED THE WORD IN ";G;" TRIES !!!"
515  CALL  - 1059: CALL  - 1059
520  GOTO 470
530  REM  DRAW GALLOWS ----------
540 FAIL = FAIL + 1
550  ON FAIL GOSUB 570,610,670,710,750,780,810
560  RETURN
570  REM  GALLOWS -----
580  HPLOT 64,159 TO 64,0 TO 144,0
590  HPLOT 120,0 TO 64,56
600  RETURN
610  REM  HEAD -----
620  HPLOT 144,0 TO 144,33 TO 147,21 TO 152,10 TO 162,9 TO 172,12 TO 174,
     24 TO 168,30 TO 156,33 TO 144,33
630  HPLOT 156,12 TO 159,15 TO 159,18 TO 153,15 TO 156,12
640  HPLOT 162,21 TO 165,21 TO 168,24 TO 165,27 TO 162,24 TO 162,21
650  HPLOT 152,20 TO 152,26 TO 154,28 TO 160,28
660  RETURN
670  REM  FIRST ARM -----
680  HPLOT 144,33 TO 132,33 TO 105,51 TO 114,75 TO 111,84
690  HPLOT 117,84 TO 114,75 TO 123,81
700  RETURN
710  REM  SECOND ARM -----
720  HPLOT 144,33 TO 156,39 TO 172,63 TO 165,87 TO 156,87
730  HPLOT 162,96 TO 165,87 TO 168,96
740  RETURN
750  REM  BODY -----
760  HPLOT 144,33 TO 138,54 TO 144,75
770  RETURN
780  REM  FIRST LEG -----
790  HPLOT 144,75 TO 114,88 TO 114,138 TO 102,132
800  RETURN
810  REM  SECOND LEG -----
820  HPLOT 144,75 TO 162,105 TO 141,135 TO 156,135
830  HCOLOR= 0
840  HPLOT 152,20 TO 152,26 TO 154,28 TO 160,28
850  HCOLOR= 7
860  HPLOT 150,22 TO 153,22 TO 158,27 TO 158,30
870  RETURN
```

我的字是什麼？與Wordlist 1



這是一個猜字比賽的古典遊戲

〃我的字是什麼？〃（What's My Word？）是一個非常古老的猜字遊戲，在維多利亞時代很受人喜愛。它是一個很受人歡迎的Master Mind 遊戲的老祖宗，筆者很喜歡它。

WORDLIST 1是一個包含153個四-字母字的順序文字檔（Sequential text file），您應該打入電腦將它SAVE在磁碟片上。將WORDLIST 1存入磁碟片的程式摘自DOS 3.3使用手册的第50頁：

```
10 REM WORDLIST1
20 T$ = "WORDLIST1"
30 REM CHR$(4) = CTRL-D
40 PRINT CHR$ (4); "OPEN ";T$
50 PRINT CHR$ (4); "WRITE ";T$
60 PRINT "ARCH"
70 PRINT "BEND"
80 PRINT "CLIP"
.....
200 PRINT "ZEST"
210 PRINT CHR$ (4); "CLOSE ";T$
220 END
```

放入WORDLIST 1的字愈多愈好，有兩個遊戲要使用到這個檔案—— 〃我

的字是什麼？″和“拼字遊戲″（Anagrams）；您當然也可以將它用在您自己設計的其他猜字遊戲裡。

“我的字是什麼？″由一項簡單的DOS 檔案處理技巧揭開序幕：140 至 200 那幾列從磁碟片讀出 WORDLIST 1，240 至250 那幾列從中隨機選出一個字作爲靶字 WRD$，然後開始進行遊戲。遊戲者的目標是設法儘快猜出那個字，不過，並不是像在“劊子手″（Hangman）遊戲中遊戲者一次只打入一個字母，而是要遊戲者每次打入一個完整的四一字母測試字（320 至 380 那幾列）。350 那一列是一個錯誤捕捉器（error trap），避免遊戲者不慎按了空白鍵和回頭鍵（return）。370 那一列允許遊戲者在絕望時按“ESCape″（脫出）鍵聲明放棄，而 690 那一列揭露謎底是什麼。

每當遊戲者猜對一個字母，而且在字中的正確位置時電腦會在螢光幕打出一個“*″，但是如果字母正確而位置不對時它會打出一個“+″（400至 510 那幾列）；這些*和＋符號可作爲下一次猜測的線索。

例如，如果電腦挑選的靶字爲HALT，而遊戲者打入TAKE，螢光幕將會顯示：

```
TYPE IN GUESS #3
TAKE * +
```

“*″是針對A的，它在HALT和TAKE兩個字中都是第二個字母；“+″是針對T的，它在兩個字中都有，但位置不同。如果遊戲者想放棄，他只須按“ESC″鍵，690 那一列便會令電腦印出謎底。

本遊戲與Master Mind 的最大差異在於謎底是一組拼成一個字的字母，不是一排隨便選擇的顏色，使得謎底與測試字有更多限制；筆者認爲這樣更有趣。謎底必須是標準的四一字母字，像QMAJ 之類的無意義字眼是不合規定的。（程

式無法檢查出這一點——因爲如果記憶中沒有預置一部包含所有可能四一字母字的巨大字典，這件事是很難辦到的。）剛開始時，您可能會認爲本遊戲根本無法玩，因爲四一字母字的數目看起來好像比四顏色可能組合的數目多得多。Master Mind 使用了六種顏色，因此四一字母組合的可能數目爲 26^4（等於 456,976）——350 多倍；不過，英語是有結構的，在四一字母字中有大部份從未被使用。事實上，整個英語字彙少於 100,000 字，四一字母字的眞正數目大約只有 2700 個——只有四顏色組合的兩倍多！四一字母組合中大約只有 1/200 爲眞正的英文字，其他的不是發不出音，就是從未被使用過。順便一提的是您想知道筆者是如何估算英語中四一字母字的數目嗎？筆者採用的是取樣法（ sampling method ）：我打開 2500 頁的牛津英文字典，隨便挑選 20 頁，發現這幾頁中總共包含了 22 個四一字母字。我將它乘上 2500／20 ，大約得到 2700 ，我因此推算整個字典中大約包含 2700 個四一字母字。

What's My Word?

```
10  REM *********************
20  REM *                   *
30  REM *   WHAT'S MY WORD?  *
40  REM *                   *
50  REM *         BY        *
60  REM *                   *
70  REM *    STUART ANSTIS  *
80  REM *                   *
90  REM *********************
100  REM
110  TEXT : HOME : VTAB 10: PRINT "ONE MOMENT, PLEASE."
120  DIM W$(10),GUESS$(20),L(200),A$(200)
130  REM  READ WORDLIST1 FROM DISKETTE-----
140 T$ = "WORDLIST1":D$ =  CHR$ (4)
150  PRINT D$;"OPEN";T$
160  PRINT D$;"READ";T$
170  FOR I = 1 TO 153
180  INPUT A$(I)
190  NEXT I
200  PRINT D$;"CLOSE";T$
210  HOME : VTAB 15: PRINT "WANT INSTRUCTIONS (Y/N) ?"
220  GET Y$
230  IF Y$ = "Y" THEN  GOSUB 720
240 K =  INT (154 *  RND (1) + 0.5)
250 WRD$ = A$(K)
260  FOR M = 1 TO 4:W$(M) =  MID$ (WRD$,M,1): NEXT M
270  HOME : PRINT "(TO GIVE UP, HIT 'ESC')"
280  VTAB 5: PRINT "TYPE IN FOUR - LETTER WORDS"
290  PRINT : PRINT " ***  DO NOT HIT RETURN !!  ***"
300 CRECT = 0:HIT = 0:G = 0
310 G = G + 1
320  PRINT : PRINT "TYPE IN GUESS # ";G;"    ";
330  IF G < 10 THEN  PRINT " ";
340  FOR J = 1 TO 4: GET GUESS$(J)
350  IF GUESS$(J) =  CHR$ (13) OR GUESS$(J) =  CHR$ (32) THEN J = J - 1: GOTO
     380: REM REJECT SPACE & CARRIAGE RETURN
360  PRINT GUESS$(J);
370  IF GUESS$(J) =  CHR$ (27) THEN  GOTO 670: REM ESC=GIVE UP
380  NEXT J
390  PRINT "  ";
400  FOR J = 1 TO 4
410  FOR K = 1 TO 4
420  IF J = K THEN  GOTO 450
430  IF GUESS$(J) = W$(K) THEN CRECT = CRECT + 1
440  GOTO 460
450  IF GUESS$(J) = W$(K) THEN HIT = HIT + 1
460  NEXT K,J
470  FOR J = 1 TO HIT + 1: IF J > 1 THEN  PRINT "*";
480  NEXT J
490  PRINT " ";
500  FOR J = 1 TO CRECT + 1: IF J > 1 THEN  PRINT "+";
510  NEXT J
520  IF HIT = 4 THEN  GOSUB 550: REM  WIN
530 CRECT = 0:HIT = 0
540  GOTO 310
550  REM  WIN -----
560  PRINT : PRINT
570  PRINT "YOU WON IN ";G;" GUESSES !!!"
580  PRINT "WANT ANOTHER GAME (Y/N) ?"
590  GET Y$
600  IF Y$ = "Y" THEN  HOME : GOTO 240
610  PRINT : PRINT
620  PRINT "TO EXIT, HIT 'ESC'"
```

待續…

```
630  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
640  GET H$
650  IF H$ =  CHR$ (27) THEN  HOME : VTAB 10: PRINT "OK. THANKS FOR PLAYI
     NG.": END
660  PRINT : PRINT  CHR$ (4);"RUN HELLO": END
670  REM  GIVE UP-----
680  PRINT : PRINT
690  PRINT "THE WORD WAS:-- ";WRD$
700  PRINT : PRINT
710  GOTO 580
720  REM  INSTRUCTIONS-----
730  HOME : VTAB 2: HTAB 15
740  PRINT "WNAT'S MY WORD ?"
750  PRINT
760  FOR M = 1 TO 8: PRINT "*****";: NEXT M
770  PRINT : PRINT "THE COMPUTER HAS THOUGHT OF A FOUR -"
780  PRINT "LETTER WORD FOR YOU TO GUESS."
790  PRINT
800  PRINT "TYPE IN A FOUR - LETTER PROBE WORD."
810  PRINT
820  PRINT "IF YOU GET A LETTER RIGHT, IN THE RIGHT PLACE, THE COMPUTER W
     ILL PRINT '*'"
830  PRINT
840  PRINT "IF YOU GET A LETTER RIGHT, BUT IN THE"
850  PRINT "WRONG PLACE, THE COMPUTER WILL PRINT '+'"
860  PRINT : PRINT : INVERSE : PRINT "  HIT ANY KEY TO CONTINUE...": NORMAL

870  GET H$
880  HOME
890  PRINT : PRINT "EXAMPLE:--"
900  PRINT : PRINT "TARGET WORD =  HALT"
910  PRINT "YOUR GUESS  =  TAKE * +"
920  PRINT : PRINT "* = A, IN THE RIGHT PLACE (2ND LETTER)"
930  PRINT "+ = T, IN THE WRONG PLACE."
940  PRINT : PRINT
950  PRINT "WHEN YOU TYPE IN A FOUR-LETTER GUESS,"
960  PRINT
970  HTAB 5: FLASH : PRINT "***";: NORMAL
980  PRINT "   DO NOT HIT RETURN !!   ";
990  FLASH : PRINT "***": NORMAL
1000  PRINT
1010  PRINT "TO GIVE UP (AT ANY TIME) AND SEE THE"
1020  PRINT "TARGET WORD, HIT 'ESC' (ON LEFT SIDE OF KEYBOARD)"
1030  PRINT : PRINT
1040  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE...   ": NORMAL
1050  GET H$
1060  RETURN
1070 K =  PEEK ( - 16384)
1080  IF K > 127 THEN  POKE  - 16368,0: PRINT K, CHR$ (K)
1090  GOTO 1070
```

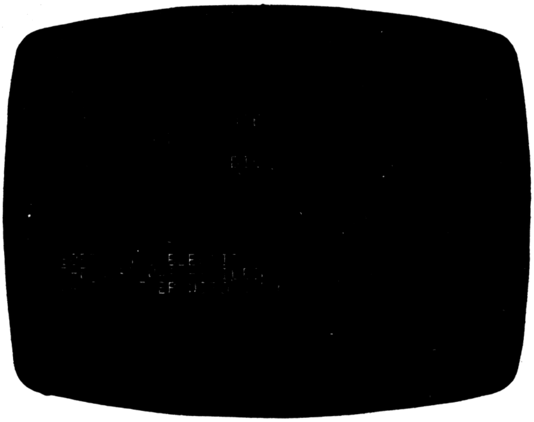
Wordlistl

ARCH	BELT	DROP	FUZZ		INTO	LENT	MELT	NAVY
ARMY	BUSH	DOZE	FLAP	GOAL	JUGS	LIVE	MOLE	NUMB
AILS	BLOW	DULL	FIST	GRIP	JOLT	LUCK	MAST	NAIL
ADZE	BOOK	DUST	FORM	GAIN	JAZZ	LOBE	MIST	OVAL
ACHE	CLAP	DARN	FUND	HAUL	JAWS	LIST	MILK	OILY
ALAS	CALM	DAME	FOAL	HEMP	KNOB	MIND	MOVE	OUST
ALSO	CROW	DISH	FLAG	HOME	KITE	MAZE	MELT	OGLE
AVID	CHOP	EVIL	FELT	HELP	KING	MOLE	MOPE	OMIT
NURN	COIL	EARN	GRAB	HIDE	LIMB	MAST	NEWT	OXEN
BEST	CURB	ELMS	GNAW	INKY	LOSS	MIST	NEXT	PASS
BALD	CLIP	EXIT	GOLD	IDEA	LAMP	MILK	NICE	POPE
BANK	DARK	FILL	GLOW	ICED	LUNG	MOVE	NAME	POST

PAGE	QUAY	ROSY	SPIN	STAB	TAPE	VEAL	WRAP	ZEST
PRIM	ROMP	RAIL	STAR	SCAN	TYPE	VILE	WAVE	ZERO
PUSH	RIND	ROBE	SWIM	TOMB	UNIT	VOID	WORN	
PILL	RAVE	SLIM	SANG	TRIP	UNDO	VOLT	WINK	
PROD	RATS	SPOT	SHIP	TAUT	VEST	WALK	WEEP	
QUIT	RAMP	SKIN	SNAP	TOPS	VAST	WIND	YOUR	

注意：當您列出這順序文字檔時它不會像上圖這樣，我們這樣印只是爲了節省篇幅。

拼字遊戲(Anagrams)



一個字雖然正確，但並非正確選擇的例子

本程式會在螢光幕上顯示出一個亂七八糟的四一字母字，遊戲者要儘快地打入正確的字。很奇怪地，在螢光幕底部時鐘嘀嗒嘀嗒響的壓力下，遊戲者常常因緊張而感到思路不暢。

拼字遊戲與前面的〝我的字是什麼?〞遊戲一樣，由磁碟片讀取名爲WORDLIST 1 的檔案揭開序幕；該檔案包含150 多個四一字母的英文字。140至220 那幾列讀入 WORDLIST 1 ，而270 至440 那幾列隨機選擇一個字，弄亂字母，再將它們顯示在螢光幕上半部。接下去，450 至540 那幾列嘀嗒嘀嗒地顯示時間，一直到遊戲者答對爲止。本程式有一項頗另人生氣的特點是它不接受其他答案；比方說如果程式從WORDLIST 1 中選出的字是CALM，它便拒絕接受CLAM 作爲正確答案。解決這項困難的唯一辦法就是確保 WORDLIST 1 中未包含諸如LIVE ，EVIL ，VILE 或VEIL 與SPOT ，POTS ，STOP 或POST之類的多重組合。將字隱藏在DATA 敘述中。

猜字遊戲選擇文字的方法是前面所述，將一個包含一長串文字的順序文字檔SAVE在磁碟片上，再由各個猜字遊戲去讀取。然而，我們當然也可以將一長串文字以DATA敍述存在程式裡，如：

```
3000 DATA ARCH, BEND, CLIP, DESK, EGGS, FONT
```

使用DATA敍述有一項缺點是把文字存在文字檔中那種方法所沒有的，那就是心術不正的遊戲者很可能事先LIST出DATA敍述偷看了所有謎底！解決的辦法是將文字以密碼形式存起來。最方便的密碼就是ASCII碼，每個字母用一個數目（A＝65，B＝66，C＝67，…）來代表。下面這個程式可爲您做編碼的事：

```
100 REM CONVERT FOUR-LETTER WORDS TO ASCII
CODE-----
110 TEXT: HOME
120 REM INPUT THE WORD-----
130 FOR J = 1 TO 4
140 GET W$ (J)
150 IF W$ (J) = CHR$ (13) THEN J = J-1: GOTO
170: REM REJECT CARRIAGE RETURNS
160 PRINT W$ (J); SPC (2);
170 PRINT
180 REM PRINT ASCII CODES-----
190 FOR J = 1 TO 4
200 PRINT ASC (W$ (J)): SPC (1);
210 NEXT J
240 PRINT : PRINT
250 GOTO 130

RUN

F  U  Z  Z
70 85 90 90

B  E  N  T
66 69 78 84

O  I  L  Y
79 73 76 89

C  R  A  M
67 82 65 77

A  B  E  T
65 66 69 84
```

```
W  I  L  D
87 73 76 68

A  R  C  H
65 82 67 72
```

假設您已經寫好了一個名爲Guess Word 的新猜字遊戲，若想將這些編了碼的字加入遊戲中，可將這些數目寫在一張紙上，或將他們印在列表機上。現在

```
LOAD GUESSWORD
```

再利用鍵盤打入

```
3000 DATA 70, 85, 90, 90, 66, 69, 78, 84, 79,
73, 76, 89, 67, 82, 65, 77, 65, 82, 67, 72,
87, 73, 76, 68, 65, 82, 67, 72
```

然後將編修過的程式再存起來：

```
SAVE GUESSWORD
```

若想讓電腦在讀取文字之前不需要知道倒底事先存了多少文字，可寫一個小迴圈一次讀入四個數目，然後利用CHR $本(J)功能將它們轉換爲字母，並且加入一列令電腦在碰到負數時停止讀資料（IF N＜0 THEN GOTO 1000 ）。最後，加入一列DATA 敍述：

```
5000 DATA -999
```

Anagrams

```
10  REM *******************
20  REM *                 *
30  REM *     ANAGRAMS    *
40  REM *                 *
50  REM *        BY       *
60  REM *                 *
70  REM *  STUART ANSTIS  *
80  REM *                 *
90  REM *******************
100  REM
110  REM
120  DIM A$(200)
130  TEXT : HOME : VTAB 10: PRINT "ONE MOMENT, PLEASE."
140  REM  LOAD WORDLIST1 FROM DISKETTE-----
150 T$ = "WORDLIST1"
160 D$ =  CHR$ (4)
170  PRINT D$;"OPEN";T$
180  PRINT D$;"READ";T$
190  FOR I = 1 TO 153
200  INPUT A$(I)
210  NEXT I
220  PRINT D$;"CLOSE";T$
230  HOME : VTAB 5: PRINT "WANT INSTRUCTIONS (Y/N) ?"
240  GET Y$
250  IF Y$ = "Y" THEN  GOSUB 880
260  REM  PICK A RANDOM TARGET WORD FROM WORDLIST1-----
270 K =  INT (153 *  RND (4) + 0.5)
280 M$ = A$(K)
290  FOR J = 1 TO 4
300 W$(J) =  MID$ (M$,J,1)
310  NEXT J
320  REM  SHUFFLE-----
330  FOR A = 1 TO 4
340 N =  INT (4 *  RND (8) + 0.5)
350  IF N = 0 THEN 340
360  FOR B = 1 TO A - 1
370 T$(A) = W$(N)
380  IF A = 1 AND B = 1 THEN 400
390  IF T$(A) = T$(B) THEN 340
400  NEXT B,A
410  HOME : VTAB 5: HTAB 18
420  FOR J = 1 TO 4
430  PRINT T$(J);
440  NEXT J
450  VTAB 20: HTAB 20: PRINT "TIME:-";
460 T = 0
470  VTAB 22: HTAB 20: PRINT  INT (10 * T) / 10;"     ";
480  FOR K = 1 TO 3
490 X =  PEEK ( - 16384)
500  IF X > 127 THEN 550
510  NEXT K
520 T = T + 0.1
530 SOUND =  PEEK ( - 16336) -  PEEK ( - 16336)
540  GOTO 470
550  POKE  - 16368,0: REM  CLEAR KEYBOARD
560 I$(1) =  CHR$ (X - 128)
570  VTAB 8: HTAB 18
580  PRINT I$(1);
590  FOR J = 2 TO 4
600  GET I$(J)
610  PRINT I$(J);
620  NEXT J
```

待續……

```
630  REM  CHECK IF GUESS IS CORRECT-----
640  FOR J = 1 TO 4
650  IF I$(J) <  > W$(J) THEN  GOTO 830
660  NEXT J
670  PRINT : PRINT
680  CALL  - 1059
690  PRINT "YOU GUESSED THE WORD IN ";T;" SECONDS !!"
700  PRINT "WANT ANOTHER WORD (Y/N) ?"
710  GET Y$
720  IF Y$ = "Y" THEN  HOME :T = 0: GOTO 270
730  HOME : VTAB 10
740  PRINT "TO EXIT, HIT 'ESC'"
750  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
760  GET K$
770  IF K$ =  CHR$ (27) THEN  GOTO 800
780  PRINT : PRINT  CHR$ (4);"RUN HELLO"
790  END
800  TEXT : HOME : VTAB 10
810  PRINT "OK. THANKS FOR PLAYING."
820  END
830  REM  WRONG GUESS-----
840  VTAB 15: HTAB 1
850  PRINT "SORRY, YOU BLEW IT."
860  PRINT "THE WORD WAS:-- ";M$
870  GOTO 700
880  REM  INSTRUCTIONS-----
890  HOME
900  VTAB 2: HTAB 16: PRINT "ANAGRAMS"
910  PRINT : PRINT
920  FOR M = 1 TO 39: PRINT "*";: NEXT M
930  PRINT : PRINT
940  PRINT "YOU WILL SEE A JUMBLED FOUR-LETTER WORD ON THE SCREEN."
950  PRINT "TYPE THE ORIGINAL WORD, AS QUICK AS YOU CAN."
960  PRINT : PRINT : PRINT
970  INVERSE
980  PRINT "  HIT ANY KEY TO CONTINUE   "
990  NORMAL
1000  GET H$
1010  RETURN
```

生日卡(Birthday Card)



這項祝賀雖有點眼花撩亂，但卻是一番心意

這張“電子生日卡”是一項在文章中設框(Windowing) 的練習。Applesoft 使用手册中並沒有好好說明文章設框的方法，但您可以在Apple Ⅱ Integer Basic 程式設計手册的第124 頁上找到，這兩種語言所用的指令是相同的。

在叫出Basic之後文章是以24 列 40 行的大小顯示在螢光幕上，當游標移到最底下一列的盡頭時整幕文章就會向上捲動。本程式可利用POKE指令讓一小塊區域捲動，而其他部份則靜止不動；那塊捲動的區域稱爲“捲動窗框”(scrolling window)，所有文字活動都將在這個窗框內進行。140 那一列包含設定窗框大小的POKE敘述您不得使用TOP或BOTTOM 作爲變數名字，因爲TOp和botTOm都包含TO在裡面。

設定捲動窗框的指令如下：

```
POKE 32, LEFTEDGE
POKE 33, WIDTH
POKE 34, TP
POKE 35, BOTTM
```

其數值通常設定於：

```
LEFTEDGE=0
WIDTH=40
TP=0
BOTTM=24
```

亦即捲動窗框正好是整個螢光幕（一幕 24 列，一列 40 個字元），以不同數值 POKE 這些位置就會改變捲動螢光幕的大小。決定好捲動窗框的大小之後 145 那一列還需要HTAB LEFTEDGE :VTAB TP將游標移入PRINT 窗框，否則程式不會正常工作；（試試看暫時將145 那一列删去會怎樣？）將145 那一列改爲："145 HOME" 也同樣可行。

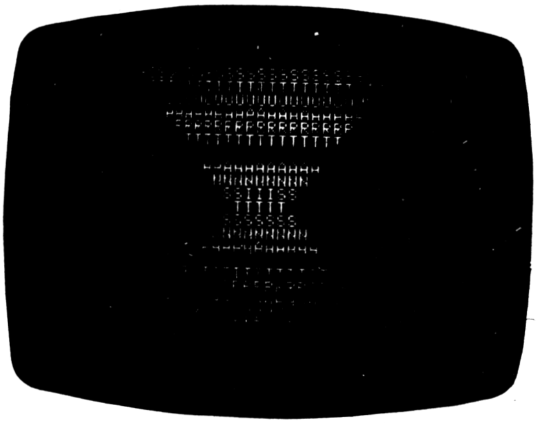
150 那一列在一個每經過一次迴圈就變小一點的文字窗框裡連續PRINT 出若干次"HAPPY BIRTHDAY,（ 您的名字）" 訊息。當它進行時那些字會呈顯出某種圖案，使螢光幕看起來好像編織品或有花邊一樣。當文字窗框縮小到看不見時它會在螢光幕上留下一些隨機，互不相關的殘渣。

您可曾想過要在編輯Applesoft 程式之前打入POKE 33 ， 33以避免將那些不必要的空白引進PRINT和DATA 敍述嗎？它會把文字窗框從正常的 40 字元寬改爲33，這樣就可除去空白。指令TEXT 可反置POKE 33， 33將窗框寬度回復爲 40 。現在，您知道爲什麼了吧。

Birthday Card

```
90  REM  BIRTHDAY CARD-----
100 B$ = "....HAPPY BIRTHDAY, "
110  TEXT : HOME
120  INPUT "PLEASE TYPE IN YOUR NAME ";W$
130 LEFTEDGE = 1:WIDTH = 40:TP = 1:BOTTM = 24: REM  SET TEXT WINDOW
140  POKE 32,LEFTEDGE: POKE 33,WIDTH: POKE 34,TP: POKE 35,BOTTM
145  HTAB LEFTEDGE: VTAB TP
150  FOR J = 1 TO 50: PRINT B$;W$;: NEXT J
160 LEFTEDGE = LEFTEDGE + 1:WIDTH = WIDTH - 2:TP = TP + 1:BOTTM = BOTTM -
    1
170  IF TP > = BOTTM THEN  TEXT : HOME : PRINT B$;W$: END
180  GOTO 140
```

貴姓大名(What's Your Name?)



這是一個很友善對待使用者的小程式

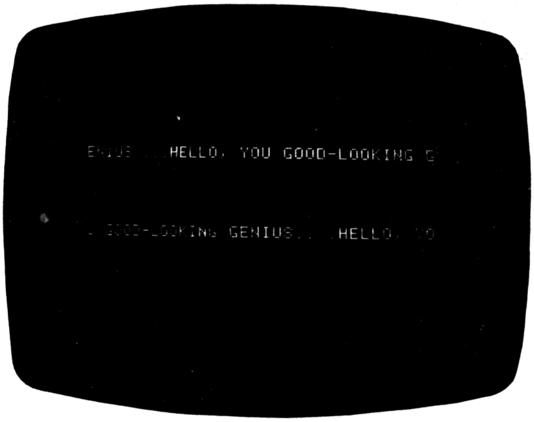
這個小程式是引導那些不習慣使用電腦而又害羞的人們進入電腦領域的好法子。遊戲者打入他的名字，然後電腦會加以處理，一次彈出一個字母將他的名字正著和倒著顯示出來。它完全是利用HTAB和VTAB完成的，您可以自己隨意做各種變化。

What's Your Name?

```
10  REM ***********************
20  REM *                     *
30  REM * WHAT'S YOUR NAME?   *
40  REM *                     *
50  REM *          BY         *
60  REM *                     *
70  REM *    STUART ANSTIS    *
80  REM *                     *
90  REM ***********************
100  REM
110  TEXT : HOME
120  PRINT "WHAT IS YOUR NAME ?"
130  PRINT : PRINT "PLEASE TYPE YOUR FULL NAME AND HIT RETURN"
140  INPUT N$
150 L =  LEN (N$)
160  IF L > 11 THEN L = 11
170  HOME
180 LM = L + 1
190  REM  PRINT LETTERS IN VERTICAL ARRAY -----
200  FOR F = 1 TO 2
210  FOR I = LM TO 1 STEP  - 1
220 H = 20 - I:V = L + 2 + (I - 2) *  SGN (F - 1.5)
230  HTAB H: VTAB V
240  FOR J = 0 TO I * 2
250  PRINT  MID$ (N$,L - I + 2,1);
260  NEXT J,I,F
270  FOR T = 1 TO 500: NEXT T
280  REM  BOUNCE LETTERS ONE AT A TIME -----
290  HOME
300 L =  LEN (N$)
305  IF L > 18 THEN L = 18
310  FOR J = 1 TO 4
320  FOR I = 1 TO L
330  VTAB 19 - I * (J > 2): HTAB 14 + I
340  PRINT  MID$ (N$,I,1);
350  VTAB 20: HTAB 14 + I
360  PRINT " ";
370  FOR T = 1 TO 100: NEXT T
380  VTAB 20: HTAB 14 + I
390  PRINT  MID$ (N$,I,1)
400  VTAB 19 - I * (J > 2): HTAB 14 + I
410  PRINT " ";
420  NEXT I,J
430  REM  POP LETTERS AT SAME LOCATION -----
440 N$ = N$ + " "
450  FOR F = 1 TO 5
460 V = 1 + 23 *  RND (1):H = 1 + 39 *  RND (4)
470  FOR I = 1 TO L + 1
480  HTAB H: VTAB V
490  PRINT  MID$ (N$,I,1);
500 X =  PEEK ( - 16336)
510  FOR T = 1 TO 200: NEXT T
```

```
520 X =  PEEK ( - 16336)
530  NEXT I
540  IF H > 38 - L THEN H = 38 - L
550  HTAB H: PRINT N$
560  NEXT F
570  FOR T = 1 TO 200: NEXT T
580  HOME
590  VTAB 16: HTAB 15
600  PRINT N$
610  FOR T = 1 TO 500: NEXT T
620  REM  RANDOM LOCATIONS -----
630  HOME
640  FOR J = 1 TO 400
650 H = 1 + (39 - L) *  RND (5)
660 V = 1 + 23 *  RND (6)
670  HTAB H: VTAB V
680  PRINT N$
690  NEXT J
700  FOR T = 1 TO 500: NEXT T
710  REM  PRINT NAME DIAGONALLY -----
720  FOR J = 1 TO 5
730  HOME
740  FOR I = 1 TO L
750 H = I:V = I: GOSUB 1090
760 H = 39 - I:V = I: GOSUB 1090
770 H = I:V = 24 - I: GOSUB 1090
780 H = 39 - I:V = 24 - I: GOSUB 1090
790 H = 20:V = I: GOSUB 1090
800 H = 20:V = 24 - I: GOSUB 1090
810 H = I:V = 12
820 H = 39 - I:V = 12: GOSUB 1090
830 H = I:V = 12: GOSUB 1090
840  NEXT I
850  FOR T = 1 TO 200: NEXT T
860  NEXT J
870  FOR V = 11 TO 13
880  HTAB 8: VTAB V
890  FOR J = 1 TO 5
900  PRINT "     ";
910  NEXT J
920  NEXT V
930  HTAB 10: VTAB 12
940  INVERSE
950  PRINT "WHAT A NICE NAME..."
960  NORMAL
970  VTAB 24
980  PRINT "WANT TO DO IT WITH ANOTHER NAME (Y/N)?"
990  GET Y$
1000  IF Y$ = "Y" THEN 110
1010  HOME : VTAB 24
1020  PRINT "HIT 'ESC' TO EXIT"
1030  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
1040  GET H$
1050  HOME
1060  IF H$ =  CHR$ (27) THEN  END
1070  PRINT : PRINT  CHR$ (4);"RUN HELLO"
1080  END
1090  REM  MARGINS SURBOUTINES -----
1100  HTAB H: VTAB V: PRINT  MID$ (N$,I,1);
1110  RETURN
```

訊息看板(Any Messages?)



這是一個捲動文字的範例

本程式可使訊息橫著“爬”過螢光幕，像塊廣告板一樣。它利用 130 和 140 那兩列中的字串處理函數使訊息同時向左和向右爬行，其原理已在第一章中說明過，如果您想要輸入自己的訊息，可將 110 那一列改爲：

```
110 INPUT "PLEASE TYPE IN A MESSAGE  ";A$
```

Any Messages?

```
100  TEXT : HOME
110 A$ = "HELLO, YOU GOOD-LOOKING GENIUS...."
120 L =  LEN (A$):B$ = A$:H = 20 - L / 2
130 A$ =  RIGHT$ (A$,L - 1) +  LEFT$ (A$,1)
140 B$ =  RIGHT$ (B$,1) +  LEFT$ (B$,L - 1)
150  VTAB 6: HTAB H: PRINT A$
160  VTAB 12: HTAB H: PRINT B$
170  IF  PEEK ( - 16384) > 127 THEN  POKE  - 16368,0: GOTO 200
180  GOTO 130
200  VTAB 20: PRINT "BYE."
```

請勿亂動(Do Not Adjust)



可開個小玩笑的程式

我們再加一個文字遊戲來胡整一下，這一次要嚇嚇您的朋友，暗中打擊一下他的士氣。當您的朋友離開房間時偸偸地在他的電腦上 RUN 這個程式，等他回來後他將會看到〝Do Not Adjust Tv〞字樣在螢光幕上閃爍，漂移，並緊張地到處亂跳，就好像他的電視機快要完蛋了。您是唬不住專家的，但是您可能使一位初學者緊張萬分…如果這是您的本意的話。

Do Not Adjust

```
100  REM  DO NOT ADJUST---
110  TEXT : HOME
120 W$(1) = "DO NOT ADJUST TV"
130 W$(2) = "                "
140 L =  LEN (W$(1))
150  REM SETUP MARGINS---
160 VT = 1:VBTM = 23: REM TOP & BOTTOM MARGINS
170 HL = 1:HR = 40 - L: REM LEFT & RIGHT MARGINS
180 V = 1 + ( RND (1) * VBTM): REM VERTICAL START POSITION
190 H = 1 + ( RND (1) * HR): REM HORIZONTAL START POSITION
200  REM PRINT---
210  VTAB V: HTAB H
220  PRINT W$(1)
230  FOR T = 0 TO 20: NEXT T
240  VTAB V: HTAB H: PRINT W$(2): REM ERASE
250  REM ALTER POSITION---
260 H = H + 2 - 4 * ( RND (1) > 0.5): REM  INCREASE OR DECREASE H
270 V = V + 1 - 2 * ( RND (1) > 0.5)
280  IF H < HL THEN H = HL
290  IF H > HR THEN H = HR
300  IF V < VT THEN V = VT
310  IF V > VBTM THEN V = VBTM
320  GOTO 210
```



蝴蝶(Butterfly)

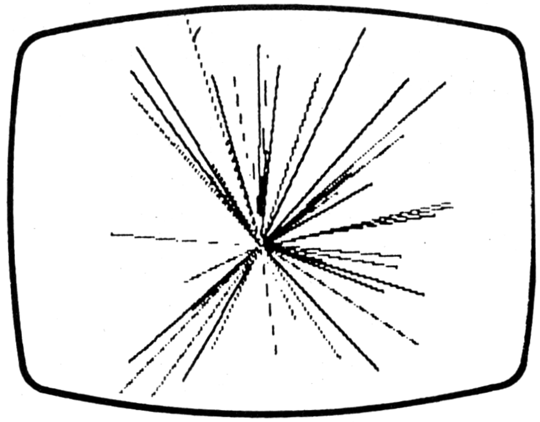
這個低解析度副程式可顯示一個以隨機路線到處飛舞的點，並在移動時隨機變換其顏色。（將本程式與高解析度〝東倒西歪〞（ Random Walk ）程式中的〝醉漢走路〞比較看看。）使用者在 60 和 70 那兩列中告訴程式希望一隻或兩隻蝴

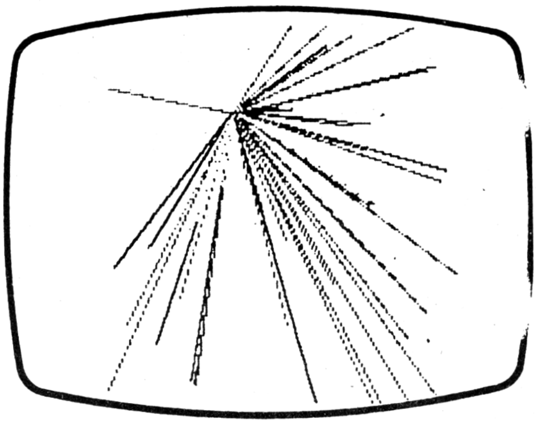
蝶；如果他希望一隻，則點在移動時不會在後面留下痕跡，因爲它每移動至一個新位置時程式便把前一個位置擦掉（ 180 那一列）。如果他希望的是兩隻，則第一隻蝴蝶仍像前面一樣擦去舊位置，但 170 那一列中的第二隻蝴蝶並不擦去舊位置，而讓痕跡留在它後面。不過，如果第一隻蝴蝶飛舞過第二隻蝴蝶的痕跡，它會在痕跡上"吃"出一個洞。（想想看爲什麼？）

Butterfly

```
10   REM    ********************
20   REM      BUTTERFLY
30   REM      BY STUART ANSTIS
40   REM    ********************
50   TEXT : HOME : VTAB 5
60   PRINT "ONE BUTTERFLY (1) OR TWO (2) "
70   GET N
80   HOME : GR
90  XO = 20:YO = 20
100  XN = XO + 2 *  RND (3) - 1
110   IF XN < 0 THEN XN = 0
120   IF XN > 34 THEN XN = 34
130  YN = YO + 2 *  RND (3) - 1
140   IF YN < 0 THEN YN = 0
150   IF YN > 39 THEN YN = 39
160   COLOR= 16 *  RND (1): PLOT XN,YN
170   IF N = 2 THEN  PLOT XN + 5,YN + 5: REM  SECOND BUTTERFLY
180   COLOR= 0: PLOT XO,YO
190  XO = XN:YO = YN
200   GOTO 100
```

烟火(Firework)





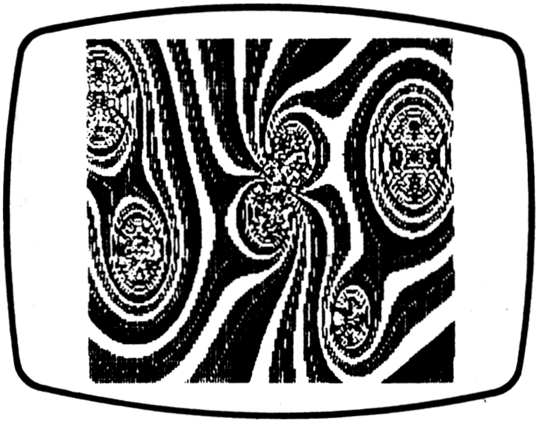
這是我們用來作爲爆炸圖形的副程式

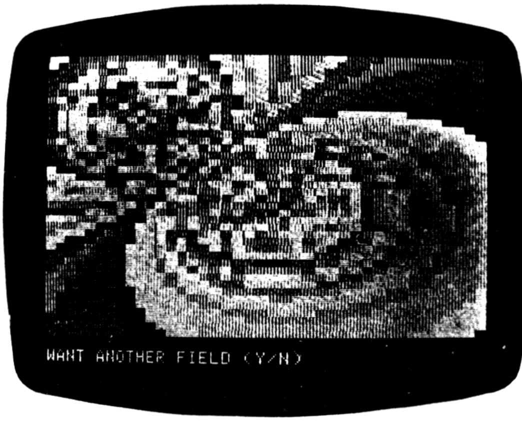
這是一個很漂亮，看起來像烟火爆炸的高解析度示範圖形，係"直昇機"（Helicopter）和"坦克"（Tank）遊戲中爆炸副程式的基礎。本程式由中央一個固定點開始，隨便選擇很多X，Y座標點，並以隨機顏色向它們畫出一大堆線而達到爆炸的效果，整個過程很簡單，但效果很好。

Firework

```
2   REM  **********************
3   REM
4   REM          FIREWORK
5   REM             BY
6   REM       STUART ANSTIS
7   REM
8   REM  **********************
10  FOR N = 1 TO 5
20  HGR2
30 J = 90 + 80 *  RND (1)
40 K = 45 + 90 *  RND (1)
50  FOR A = 1 TO 200 *  RND (1)
60 HUE =  INT (7 *  RND (1))
70  HCOLOR= HUE
80 X = 280 *  RND (1)
90 Y = 190 *  RND (1)
100  HPLOT J,K
110  HPLOT  TO X,Y
120  NEXT A
130  NEXT N
140  FOR L = 1 TO 50
150  PRINT
160  NEXT L
170  TEXT
180  END
```

高解析度與低解析度等電位圖 ※





高解析度等電位圖（上圖）與低解析度等電位圖（下圖）

這是一對繪圖副程式，一個是低解析度的，一個是高解析度的；它們是根據一本由Thomas Dwyer和Margot Critchfield合著的BASIC and Personal Computer 中的一個程式來繪出彩色等電位圖的。

任何一個帶有電荷的物體都會在它四週產生一個電場，其電位（potential）與距離成反比。如果同時有幾個帶電荷物體存在，則電場中任意一點的電位等於各個物體所感應電位的總和；例如，如果有一點距離帶著C1、C2和C3 電量的

物體分別爲D1、D2和D3，則該點的總電位等於 $C1/D1 + C2/D2 + C3/D3$。

程式會要求您指定每個電荷的位置和電量；注意：您放入的電荷愈多程式RUN的時間就愈長。程式計算出每一點的淨電位，然後像描繪地圖的等高線一樣描繪出等電位線，並以顏色來代表電位的大小。首先試試低解析度程式，它大約需要5至20分鐘；高解析度圖形看起來很漂亮，但是它大約需要花費1至8個小時才能畫好，因此最好在晚上畫。您可以設法照張彩色照片留作紀念。

※譯者註：原著爲描繪電場圖，但譯者將它改爲描繪等電位圖，因爲原作者誤將電位以爲電場强度。蓋空間上某一點之電場强度爲向量，其强度與距離平方成反比，而電位係純量，其大小與距離成反比。

Electric Hi-res

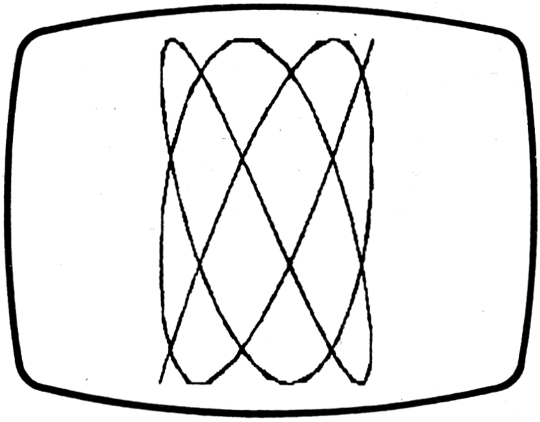
```
10   REM  *********************
20   REM      ELECTRIC FIELDS
30   REM            BY
40   REM  T.DWYER & M.CRITCHFIELD
50   REM         GRAPHICS BY
60   REM       STUART ANSTIS
70   REM  *********************
80   TEXT : HOME
90   PRINT "ELECTRIC FIELDS PLOTTED IN HIRES"
100   PRINT "AFTF° DWYER & CRITCHFIELD"
110   PRINT : PRINT "***  WARNING --- THIS PROGRAM TAKES"
120   FLASH : PRINT "HOURS";: NORMAL : PRINT " TO PLOT !!": PRINT
130   PRINT "(RUN IT OVERNIGHT?)"
140   PRINT : PRINT
150   PRINT "HOW MANY CHARGES (1--9) ?"
·60   INPUT N$:N =  VAL (N$): IF N < 1 OR N > 9 THEN  GOTO 90
170   FOR I = 1 TO N
180   PRINT "TYPE IN X & Y POSITION OF CHARGE #";I
190   INPUT "X (0-279)...";A(I)
200   INPUT "Y (0-191)...";B(I)
210   IF A(I) > 279 OR B(I) > 191 THEN  CALL  - 1059: GOTO 180
220   NEXT I
230   PRINT : PRINT
240   PRINT "TYPE IN THE SIZE OF EACH CHARGE"
250   PRINT "(-3 TO +3)"
260   FOR I = 1 TO N: PRINT "CHARGE ";I;"...";: INPUT C(I): NEXT I
270   PRINT : PRINT : PRINT : PRINT : PRINT
280   HGR
290   GOTO 410
```

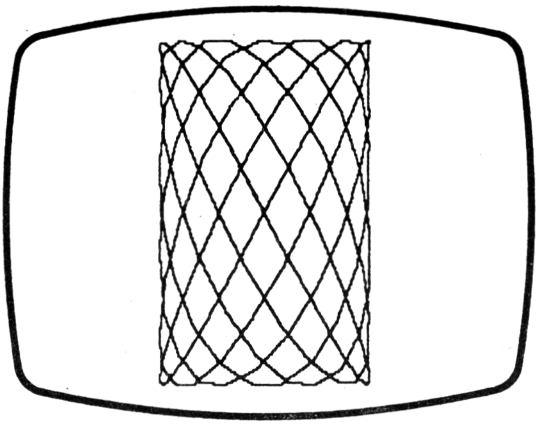
```
300  REM  CALCULATE FIELD STRENGTH ----
310 V = 0
320  FOR I = 1 TO N
330 R =  SQR ((X - A(I)) ^ 2 + (Y - B(I)) ^ 2)
340  IF R = 0 THEN 360
350 V = V + C(I) / R: GOTO 370
360 V = 99999: GOTO 380
370  NEXT I
380  RETURN
390  STOP
400  REM  PLOTTING SUBROUTINE -----
410  FOR X = 1 TO 279
420  FOR Y = 1 TO 159
430  GOSUB 310
440 Z = V
450 Z = (Z + 1) * 160
460 Z = 7 * (Z / 7 -  INT (Z / 7))
470  HCOLOR= Z
480  HPLOT X,Y
490  NEXT Y
500  NEXT X
510  END
```

Electric Lo-res

```
10  REM  *********************
20  REM      ELECTRIC FIELDS
30  REM            BY
40  REM    T.DWYER & M.CRITCHFIELD
50  REM          GRAPHICS BY
60  REM       STUART ANSTIS
70  REM  *********************
80  TEXT : HOME : VTAB 3
90  PRINT "HOW MANY CHARGES (1--9) ?"
100  INPUT N$:N =  VAL (N$): IF N < 1 OR N > 9 THEN  GOTO 90
110  FOR I = 1 TO N
120  PRINT "TYPE IN THE X AND Y POSITIONS (0--39)"
130  PRINT "OF CHARGE ";I;"...";
140  INPUT A(I),B(I)
150  NEXT I
160  PRINT
170  PRINT "TYPE IN THE SIZE OF EACH CHARGE (-3 TO +3)..."
180  FOR I = 1 TO N: PRINT "CHARGE ";I;"...";: INPUT C(I): NEXT I
190  PRINT : PRINT : PRINT : PRINT : PRINT
200  GR
210  GOTO 330
220  REM  CALCULATE FIELD STRENGTH ----
230 V = 0
240  FOR I = 1 TO N
250 R =  SQR ((X - A(I)) ^ 2 + (Y - B(I)) ^ 2)
260  IF R = 0 THEN 280
270 V = V + C(I) / R: GOTO 290
280 V = 99999: GOTO 300
290  NEXT I
300  RETURN
310  STOP
320  REM  PLOTTING SUBROUTINE -----
330  FOR X = 0 TO 39
340  FOR Y = 0 TO 39
350  GOSUB 230
360 Z = V
370 Z = (Z + 1) * 100
380 Z = 16 * (Z / 16 -  INT (Z / 16))
390  COLOR= Z
400  PLOT X,Y
410  NEXT Y
420  NEXT X
430  VTAB 22
440  PRINT "WANT ANOTHER FIELD (Y/N) ?"
450  GET Y$
460  IF Y$ = "Y" THEN 80
470  PRINT "WANT TO RETURN TO MENU (Y/N) ?"
480  PRINT
490  GET M$
500  IF M$ = "Y" THEN  TEXT : HOME : PRINT : PRINT  CHR$ (4);"RUN HELLO"
510  END
```

李沙育圖(Lissajou)





這兩個例子顯示不同的諧波比

鐘擺在擺動時是行簡諧（ simple harmonic motion ），其擺動路徑若在時間軸上描繪下來又稱為正弦波（ sine wave ）。假想有一支長鐘擺正南北向擺動著，而其尖端又連著一支短鐘擺東西向擺動著，則下方鐘擺擺動的路徑可描繪出一種稱為李沙育圖（ Lissajou figure ）的 8 一 字圖。電子工程師只要將兩種不同頻率的正弦波信號加到示波器的 X 與 Y 輸入端就可得到李沙育圖，這種圖形控制示波器螢光幕上光點的垂直與水平移動。

您可以藉輸入兩種不同頻率H（水平）與V（垂直）來實驗李沙育圖，這兩個正弦波是在130至140那兩列中產生的。若想得到一幅自動，永不停止的環狀圖形顯示，可做下列修改：

```
DEL 20, 40
DEL 180, 290
50 H = INT (10*RND(1)): V = INT (10 * RND
(2))
180 GOTO 50
```

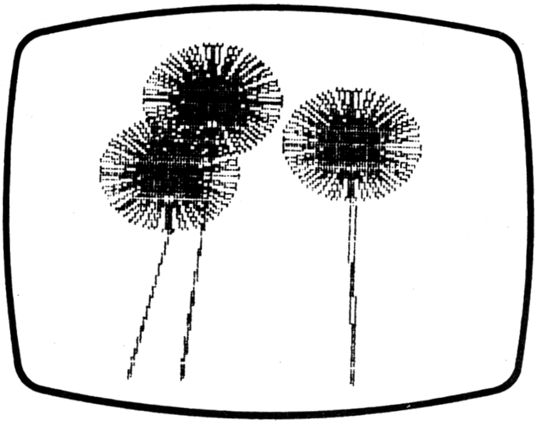
所有李沙育圖中最簡單的是一個圓，這是當一個點的垂直運動為正弦波，而其水平運動為餘弦波（ cosine　wave ） 時所產生之軌跡；餘弦波其實只是相位移動了 90 度的正弦波。X，Y座標之計算方法如下：

```
X = CENTER + RADIUS * SIN (ANGLE)
Y = CENTER + RADIUS * COS (ANGLE)
```

Lissajou

```
20   TEXT : HOME
30   HTAB 10: PRINT "LISSAJOU FIGURES"
40   VTAB 10: HTAB 5
50   INPUT "HARMONIC RATIO: TYPE IN TWO NUMBERS AND HIT RETURN AFTER EACH
       ... ";H,V
100   HGR
110   HCOLOR= 3
115   HPLOT 140,80
120   FOR K = 0 TO 6.4 STEP 0.05
130  X = 140 + 80 *  SIN (H * K)
140  Y = 80 + 80 *  SIN (V * K)
160   HPLOT  TO X,Y
170   NEXT K
180   VTAB 24
190   PRINT "WANT ANOTHER (Y/N)    ";
200   GET Y$
210   IF Y$ = "Y" THEN 20
220   TEXT : HOME : VTAB 10
230   PRINT "HIT 'ESC' TO EXIT"
240   PRINT : PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
250   GET H$
260   IF  ASC (H$) = 27 THEN  END
270   PRINT
280   PRINT  CHR$ (4);"RUN HELLO"
290   END
```

蒲公英(Dandelions)



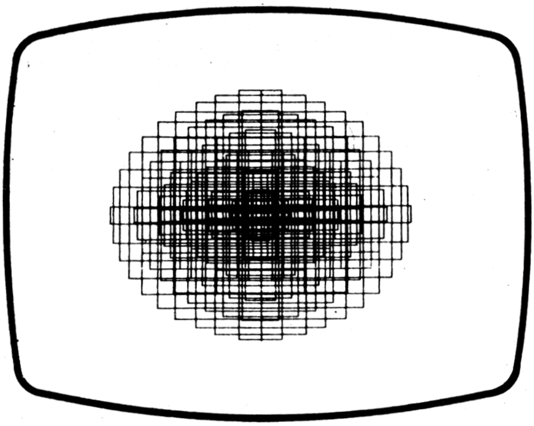
簡單又美麗的圖案

這個簡單程式可畫出三株蒲公英，每株蒲公英包含許多由一個圓心射出來之輻射線，像盛開的烟火一樣。

Dandelions

```
10  REM ********************
20  REM    DANDELIONS
30  REM ********************
40  HOME : HGR
50 R = 25
60  FOR FLOWER = 1 TO 3
70  HCOLOR= 3
80 X = 30 + 210 * RND (1)
90 Y = 30 + 50 * RND (2)
100  FOR A = 0 TO 6.3 STEP 0.1
110 XA = X + R * SIN (A)
120 YA = Y + R * COS (A)
130  HPLOT X,Y TO XA,YA
140  NEXT A
150  HCOLOR= 1
160 RAN = 30 * RND (3) - 30 * RND (4)
170  HPLOT X,Y + R TO X + RAN + 1,159 TO X + RAN + 1,159 TO X + 1,Y + R
180  NEXT FLOWER
190  VTAB 24
```

酒杯墊(Doily)



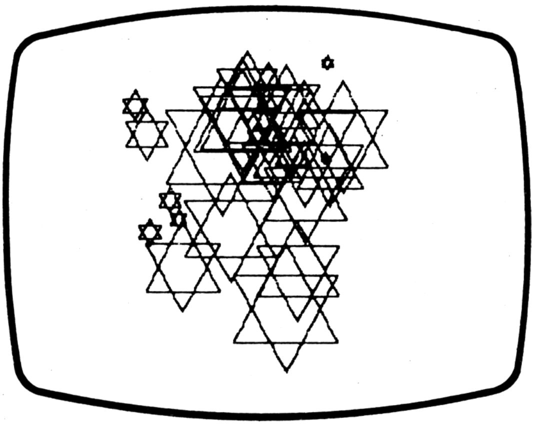
這麼複雜的設計其實只是由一連串長方形重疊而成

這個程式可畫出一連串長方形，先畫矮胖的，再畫瘦長的，它們的四個角都內接於同一個圓上，而這個外接圓並漸漸由內向外擴大，其圖案有點像英國午茶派對中用來墊在蛋糕或酒杯下面打了孔的紙墊。

Doily

```
50  REM  DOILY-----
60 XC = 140:YC = 96
70 PI = 3.14159
90  HGR2 : HCOLOR= 3
95  FOR R = 20 TO 95 STEP 15
100  FOR A = 0 TO PI STEP 0.3
110 DX = R *  SIN (A)
120 DY = R *  COS (A)
130  GOSUB 300
140  NEXT A,R
280  GOTO 90
300 X1 = XC - DX:X2 = XC + DX
310 Y1 = YC - DY:Y2 = YC + DY
320  HPLOT X1,Y1 TO X2,Y1 TO X2,Y2 TO X1,Y2 TO X1,Y1
330  RETURN
```

六角星圖(Stars)



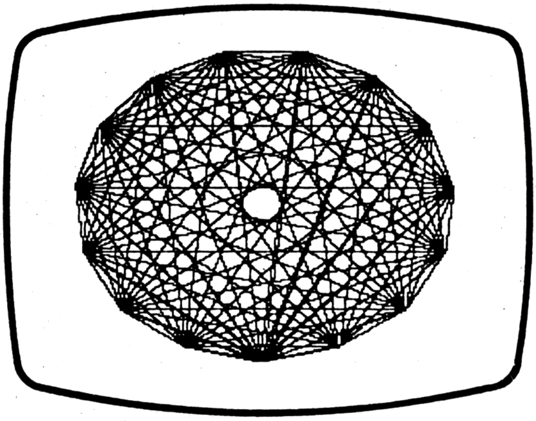
每個星星的位置與大小是隨機決定的

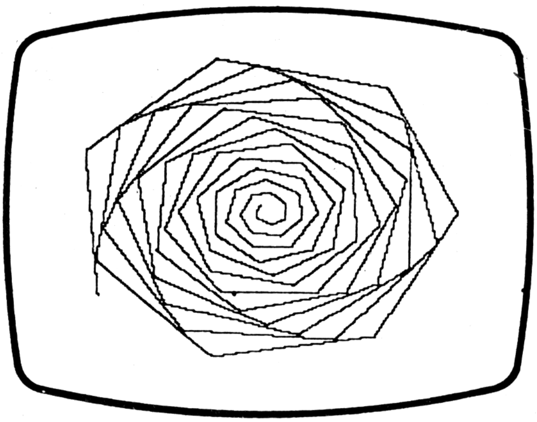
本程式可將一連串高解析度，各種大小的六角星星隨機畫在螢光幕上。您可以當做一種練習，利用下一個程式〝多邊形與旋形圖〞（ Polygons and Squirals ）中所用的多邊形畫法改寫這個程式。

Stars

```
10  REM STARS-----
20  HGR2
30 HC =  INT (8 *  RND (5))
40  IF HC = 0 OR HC = 4 THEN  GOTO 30
50  HCOLOR= HC
60 X = 50 + 180 *  RND (1): REM  POSITION
70 Y = 30 + 90 *  RND (2)
80 W = 50 *  RND (3): REM  SIZE
90 Z = 0.866 * W
100  HPLOT X,Y TO X + Z,Y + W TO X - Z,Y + W TO X,Y
110  HPLOT X + Z,Y + W / 3 TO X,Y + 3 * Z / 2 TO X - Z,Y + W / 3 TO X + Z
     ,Y + W / 3
120  IF X > 220 THEN  HGR2 : REM  CLEAR SCREEN OCCASIONALLY
130  GOTO 30
```

多邊形與旋形圖(Polygons and Squirals)





這是兩個以幾何爲基礎的顯示程式

這些是自動，永不停止的迴圈式（ looping ）顯示程式，這兩個程式都是利用畫圓的方法產生正規的多邊形。旋形圖（ Squirals ）*原先是以 Seymour Papert 的 Logo 語言寫成的，但經過 Fred Gunther 以 Basic 改寫後在 1983年元月份的 Creative Computing 雜誌發表的。多邊形與旋形圖的描繪原理在第一章中都討論過。

---

* 暫譯

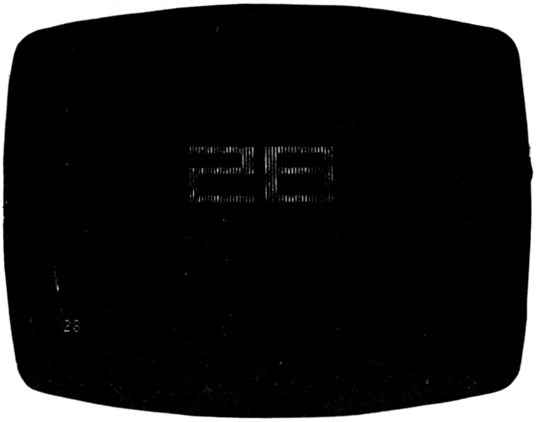
Polygons

```
10  REM POLYGON-----
20  REM SET THE VARIABLES-----
30 PI = 3.14159
40 XC = 140:YC = 96
50 R = YC - 3
60  TEXT : HOME
70  DIM X(50),Y(50)
80  FOR N = 3 TO 20: REM  N = NUMBER OF SIDES
90 J = 0:S = 2 * PI / N
100  HGR2 : HCOLOR= 3
110  REM CALCULATE CO-ORDINATES-----
120  FOR A = 0 TO 2 * PI + S STEP S
130 J = J + 1
140 X(J) = XC + R *  SIN (A)
150 Y(J) = YC + R *  COS (A)
160  NEXT A
170 JMAX = J
180  REM PLOT THE FIGURE-----
190  FOR J = 1 TO JMAX
200  FOR K = 1 TO J
210  HPLOT X(J),Y(J) TO X(K),Y(K)
220  NEXT K,J,N
```

Squirals

```
10  REM   SQUIRALS-----
20  REM BY FRED GUNTHER, CREATIVE COMPUTING JANUARY 1983---
30 PI = 3.14159:W = PI / 180
50 ANGLE = 360 *  RND (3):R = 5 *  RND (3): REM  RADIUS
60  HGR2 :X = 140:Y = 96: HCOLOR= 3:A = ANGLE
70 XN = X + R *  COS (W * A):YN = Y + R *  SIN (W * A)
80  ONERR  GOTO 50
90  HPLOT X,Y TO XN,YN
100 A = A + ANGLE
110 R = R + 1
120 X = XN:Y = YN
130  GOTO 70
```

碼錶(Stopwatch)



這個副程式可用於任何需要計時的遊戲中

假設您有一個在遊戲期間需要計時的遊戲，您可能會想要在螢光幕上顯示一個數字鐘，下面就是一個在TEXT模式中進行的最簡易方法：

```
10 S = - 16336
20 N = 0
30 HOME
40 TICK = PEEK (S) - PEEK (S)
50 VTAB 21: HTAB 20: PRINT N
60 FOR PAUSE = 1 TO 560: NEXT PAUSE
70 N = N + 1
80 GOTO 30
```

這個副程式可在螢光幕底部中央顯示出所經過的秒數（將螢光幕的其餘部份留作繪圖用）。N為所經過的秒數，在50那一列中PRINT出來，而 60那一列爲計時迴圈，提供相當準確的一秒計時（試試看讓程式計數一、二分鐘）。 70那一列將N加1，而 80 那一列跳回去重頭開始。

這個程式用起來還可以，但〝碼錶〞程式可用低解析度顯示數字而產生更佳效果，它的兩位數分別存放在變數TENS和UNITS中 並在40至80那幾列中增

量及印出來。這些數字另外藉 900 至 1900 那幾列中的副程式以低解析度描繪出來，而各個數字副程式又是藉呼叫一組位於 3000 至 3070 那幾列中的〝七段〞副程式完成的。這完全是採用計算器與數字錶中利用七段發光二極體（ LED ）來湊出數字的同樣原理，比利用一塊記憶體來儲存各個數字的方法簡短且更容易處理。

計時迴圈位於 140 那一列中，〝IF HUE>0〞條件表示數字一定是在計時期間才顯示出來，以避免閃爍。計時迴圈只計數至 465 （不是前面的 560 ），那是因爲描繪數字需要一些額外時間。不要在此處要求秒鐘很精確，因爲描繪像 6 的五段數字比像 1 的一段數字所需的時間長；不過，您可以發現這時鐘在大多數場合中已有足夠的準確度。您也可以把計時迴圈置於遊戲控制器的控制之下，只要將 140 那一列改爲：

```
140 IF HUE > 0 THEN FOR PAUSE = 0 TO PDL (0):
NEXT PAUSE
```

如果您將〝碼錶〞程式納入其他遊戲中，您可以利用下面方法同時監視鍵盤：

```
FOR PAUSE = 0 TO 100: IF PEEK (-16384) >127
THEN GOSUB 1000: NEXT PAUSE
```

假若您想把數字放在螢光幕的角上，將螢光幕其餘部份騰出來做低解析度遊戲，這是很容易的。 30 那一列指定右邊數字的寬度和位置（列和行），您只要改變這些數值即可。

本程式只能在螢光幕上描繪兩位數，因此它計數至 99 後便會由零重新開始。您可以當作一個練習，將程式加以修改使它能在重新循環之前計數至999 。

Stopwatch

```
10  REM DIGITAL STOPWATCH-----
20  TEXT : HOME : GR
30 W = 5:R = 15:C = 20: REM  DIGIT SIZE & POSITION
40 UNITS =  - 1:TENS = 0
50 UNITS = UNITS + 1
60  IF UNITS > 9 THEN UNITS = 0:TENS = TENS + 1
70  IF TENS > 9 THEN 40: REM ON99 RESET TO ZERO
80  HOME : PRINT TENS;UNITS
90 CC = 15 *  RND (1) + 1
100  FOR HUE = CC TO 0 STEP  - CC
110  COLOR= HUE
120  ON UNITS + 1 GOSUB 1000,1100,1200,1300,1400,1500,1600,1700,1800,1900

130  IF TENS > 0 THEN  GOSUB 170
140  IF HUE > 0 THEN  FOR PAUSE = 0 TO 465: NEXT PAUSE: REM  TIMING LOOP

150  NEXT HUE
160  GOTO 50
170  REM NUMBER SUBROUTINES CHOOSE SEGMENTS-----
180 C = C - W - 2: ON TENS + 1 GOSUB 1000,1100,1200,1300,1400,1500,1600,1
    700,1800,1900:C = C + W + 2: RETURN
900  REM NUMBERS-----
990  REM ZERO---
1000  GOSUB 3010: GOSUB 3030: GOSUB 3040: GOSUB 3050: GOSUB 3060: GOSUB 3
    070: RETURN
1090  REM ONE---
1100  GOSUB 3060: GOSUB 3070: RETURN
1190  REM TWO---
1200  GOSUB 3010: GOSUB 3020: GOSUB 3030: GOSUB 3050: GOSUB 3060: RETURN

1290  REM THREE---
1300  GOSUB 3010: GOSUB 3020: GOSUB 3030: GOSUB 3060: GOSUB 3070: RETURN

1390  REM FOUR---
1400  GOSUB 3020: GOSUB 3040: GOSUB 3060: GOSUB 3070: RETURN
1490  REM FIVE---
1500  GOSUB 3010: GOSUB 3020: GOSUB 3030: GOSUB 3040: GOSUB 3070: RETURN

1590  REM SIX---
1600  GOSUB 1500: GOSUB 3050: RETURN
1690  REM SEVEN---
1700  GOSUB 3010: GOSUB 3060: GOSUB 3070: RETURN
1790  REM EIGHT---
1800  GOSUB 1000: GOSUB 3020: RETURN
1890  REM NINE---
1900  GOSUB 1300: GOSUB 3040: RETURN
3000  REM PLOT SEGMENTS-----
3010  HLIN C,C + W AT R: RETURN
3020  HLIN C,C + W AT R + 3: RETURN
3030  HLIN C,C + W AT R + 7: RETURN
3040  VLIN R,R + 3 AT C: RETURN
3050  VLIN R + 3,R + 7 AT C: RETURN
3060  VLIN R,R + 3 AT C + W: RETURN
3070  VLIN R + 3,R + 7 AT C + W: RETURN
```

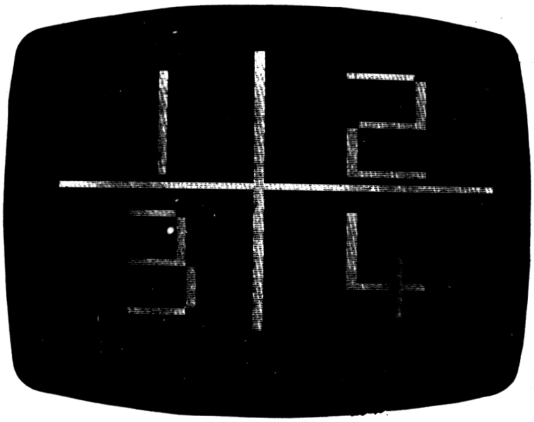


記憶人(Memory Man)

本程式模擬一個有四種顏色的面板，它以困難程度遞增的順序發出不同的顏色與音調，測驗您對光和聲音的記憶力。

在模擬當中會有四個彩色象限（Quadrants）出現在螢光幕上，它們分別標著1、2、3和4，其中有一個象限會發亮並發出一個音調，您必須打入正確的象限號碼；接下去兩個象限發亮，然後三個，…，以此類推。您每一回合的工作

就是打出您剛剛看到的順序。每個象限永遠伴隨著相同的音調，您每答對一次電腦就會在前面所出現的順序之後再加一個新象限和音調。



讓電腦來測驗您的記憶力

四個彩色象限是以低解析度圖形畫在螢光幕上，而音調是由一個摘錄自 John Barnes 和 Stan Mathes 在 1980 年元月份的 Rainbow Apple Users' Newsletter 上發表的機器語言副程式產生的。

本遊戲的玩法說明放在 530 至 690 那幾列的副程式中，而 700 至 790 那幾列則描繪象限的分界線。象限與音調出現的隨機順序是在遊戲開始之前就存在一個陣列中（150 至 180 那幾列），並由 190 至 300 那幾列呈現出來。280 那一列呼叫伴隨著象限一齊出現的音調副程式。遊戲者回答的順序由 310 至 360 那幾列收集；如果它是正確的，440 之前的那幾列會奏出一小段音樂，然後發出下一個序列。如果遊戲者錯了，800 至 920 那幾列會告訴他對了幾次，然後令程式重新開始。

Memory Man

```
10  REM *******************
20  REM *    MEMORY MAN   *
30  REM *        BY       *
40  REM *  STUART ANSTIS  *
50  REM *******************
60  REM
70  REM
75  TEXT : HOME : VTAB 10
80  PRINT "WANT INSTRUCTIONS (Y/N) ?"
90  GET Y$
100  IF Y$ = "Y" THEN  GOSUB 520: REM  INSTRUCTIONS
110  GOSUB 1000: REM  TONES SUBROUTINE ---
120  DIM S(20),R(20)
130  TEXT : HOME : GR
140  COLOR= 15: HLIN 0,39 AT 19: VLIN 0,39 AT 19
145  GOSUB 1100: REM  PLOT DIGITS
150  REM   FILL ARRAY BEFORE GAME STARTS -----
160  FOR I = 1 TO 20
170 S(I) =  INT (4 *  RND (1)) + 1
180  NEXT I
190  REM  PRESENT THE SEQUENCE -----
200  VTAB 21: HTAB 5: PRINT "1 2"
210  VTAB 22: HTAB 5: PRINT "3 4"
215  COLOR= 0: GOSUB 1100: REM  ERASE DIGITS
220  FOR I = 1 TO N
230 A = 2 * (S(I) / 2 -  INT (S(I) / 2))
240 B =  INT (S(I) / 2 - 0.5)
250 X = 20 - 20 * A
260 Y = 20 * B
270 D = 50
280  ON S(I) GOSUB 940,950,960,970
290  GOSUB 700
300  NEXT I
305  COLOR= 15: GOSUB 1100: REM  RE-PLOT DIGITS
310  REM  COLLECT PLAYER'S RESPONSES -----
320  FOR I = 1 TO N
330  GET R(I)
340  IF R(I) <  > S(I) THEN  GOTO 800: REM   MISTAKE ---
350  NEXT I
360 N = N + 1
370  PRINT "CORRECT !!!"
380  REM  TUNE FOR CORRECT RESPONSE ---
390  FOR K = 1 TO 2
400  FOR J = 150 TO 50 STEP  - 10
```

```
410 P = J:D = 6
420  GOSUB 980
430  NEXT J,K
440  HOME
450  GOTO 200: REM   CORRECT, SO PRESENT NEXT TRIAL ---
460  REM  CLEAR ARRAY FOR NEXT GAME ---
470  FOR J = 1 TO 5: GOSUB 730: NEXT J
480  FOR I = 1 TO 20
490 R(I) = 0:S(I) = 0
500  NEXT I
510  RETURN
520  REM  INSTRUCTIONS -----
530  PRINT  TAB( 15);"MEMORY MAN"
540  PRINT
550  PRINT " PUT THE FINGERS OF YOUR LEFT HAND"
555  PRINT "COMFORTABLY ON KEYS 1,2,3,4 IN THE TOP"
570  PRINT "ROW. ON THE TV SCREEN YOU WILL SEE FOUR"
575  PRINT "COLORED QUADRANTS, ARRANGED LIKE THIS:": PRINT : PRINT
580  PRINT  TAB( 15);"1 2": PRINT  TAB( 15);"3 4"
590  PRINT
600  PRINT "THEY WILL FLASH IN SEQUENCE.  WATCH THE SEQUENCE CAREFULLY AN
     D REMEMBER IT."
610  PRINT "NOW TAP OUT THE SAME SEQUENCE ON THE"
613  PRINT "KEYBOARD. IF YOU GET IT RIGHT, THE"
615  PRINT "COMPUTER WILL SAY SO, AND ADD ONE MORE"
617  PRINT "TO THE SEQUENCE."
620  PRINT "YOU CAN CONTROL PRESENTATION SPEED WITH PADDLE # ZERO"
625  PRINT
630  PRINT "    ***   HAVE FUN  !!!   ***"
640  PRINT
650  INVERSE
660  PRINT " HIT SPACE BAR TO START THE GAME "
670  NORMAL
680  GET K$
690  RETURN
700  REM  PLOT SQUARE -----
710  FOR K = S(I) TO 0 STEP  - S(I)
720  COLOR= K
730  FOR J = 0 TO 18
740  HLIN X,X + 18 AT Y + J
750  NEXT J
760 PAUSE = 255 -  PDL (0)
770  FOR TIME = 1 TO PAUSE: NEXT TIME
780  NEXT K
790  RETURN
800  REM  MISTAKE -----
805  PRINT : PRINT
810  PRINT "YOU GOT ";N - 1;" CORRECT "
820  FOR J = 200 TO 255
830 P = J:D = 6
840  GOSUB 980
850  NEXT J
860 N = 1
870  FOR TIME = 1 TO 500: NEXT TIME
875  PRINT : PRINT : PRINT
880  PRINT "HERE'S ANOTHER RUN."
890  FOR TIME = 1 TO 500: NEXT TIME
900  PRINT : PRINT : PRINT
910  GOSUB 480
920  GOTO 130: REM   NEW SEQUENCE
930  REM  TONES FOR EACH QUADRANT -----
940 P = 193: GOSUB 980: RETURN
950 P = 153: GOSUB 980: RETURN
960 P = 129: GOSUB 980: RETURN
970 P = 115: GOSUB 980: RETURN
980  POKE 768,P: POKE 769,D: CALL 770: RETURN
990  REM  TONES SUBROUTINE -----
1000  POKE 770,173: POKE 771,48: POKE 772,192: POKE 773,136: POKE 774,208
     : POKE 775,5: POKE 776,206: POKE 777,1: POKE 778,3: POKE 779,240: POKE
     780,9: POKE 781,202
1010  POKE 782,208: POKE 783,245: POKE 784,174: POKE 785,0: POKE 786,3: POKE
     787,76: POKE 788,2: POKE 789,3: POKE 790,96: RETURN
1100  REM  PLOT DIGITS -----
1110  VLIN 3,17 AT 10
1120  HLIN 27,32 AT 3: HLIN 28,33 AT 10: HLIN 27,33 AT 17
1130  VLIN 4,9 AT 33: VLIN 11,17 AT 27
1140  HLIN 7,11 AT 23: HLIN 7,12 AT 30: HLIN 7,12 AT 37
1150  VLIN 31,36 AT 13
1160  VLIN 24,30 AT 12
1170  VLIN 23,33 AT 27
1180  VLIN 29,37 AT 31
1190  HLIN 27,33 AT 33
1195  RETURN
```

風琴(Organ)

注意：這是本書唯一以 Integer Basic 寫成的程式，並不適用於Applesoft，因爲本程式實際上是一個充滿像〝GOSUB（2*X－Y）〞之類計算方式的GOSUB敍述的對照表（ lookup table ）。很不幸地，Applesoft 並不允許計算方式的GOSUB或GOTO敍述，而完全依賴較笨的ON X GOSUB 100，200，300……格式，要想將本程式改寫爲這種格式是很困難的；因此，千萬別把它打入您的Apple，除非您擁有 Integer Basic！（下一節的〝音樂〞程式是Applesoft 程式，您可以使用它。）

本程式可使您的Apple變爲一台風琴，利用鍵盤上的字鍵作爲琴鍵，按下那些鍵可產生某種音樂。

音符是藉傳送脈衝至Apple的揚聲器而產生的。利用Basic敍述X＝PEEK（－16336）產生一個單音是很簡單的，但它只能產生一個單音，或者頂多由一連串單音組成蜂鳴聲。Basic本身不夠快，無法在一定時間內要揚聲器以正常時間間隔彈奏出1秒100至10,000赫的各種樂音。變數P（音高）固定脈衝之間的時間間隔而決定某個全音的頻率，而D（拍長）固定該全音延續時間的長短。P和D的數值都在0與255之間；對D而言，0最短，255最長，而對P而言，0最高，255最低。

32000與32005那幾列內的副程式將一組機器語言敍述POKE到0至20的記憶位置內。揚聲器受位置 $02內的機器語言碼刺激發聲，然後延遲一段由位置 $00內所定的時間(P)。（注意：記憶位置通常以十六進位（以 16 爲底）表示，不是十進位（以 10 爲底）。十六進位數目前面一定冠上一個金錢符號。）

這個過程一直重複到該全音延續一段由D（ $01位置內的數值）所決定的時間間隔；程式於焉展開（ $14敍述）。本副程式是由Apple參考手册第一版的第 45 頁摘錄來的，現在已很難找到。注意，Basic 用來指定音高和拍長還是夠快的，因爲每一音符只需做一次，產生各個全音的工作則交給速度很快的機器語言副程式。

這個副程式只需在 10 那一列中呼叫一次，從那之後，該機器語言副程式便存在記憶體內，然後在每次要發出一個音符時藉 40 那一列中的 CALL 2 敍述呼叫它；CALL 2 的意思就是〝呼叫由記憶位置 $02開始的副程式〞。

本程式的主迴圈在20 至126 那幾列中，30 那一列檢查是否有鍵被按下；如果有，100 那一列便清除鍵盤。110 那一列爲本程式的心臟，當有鍵被按下時程式會指定一個介於 129 與 255 之間的字鍵代碼給變數 KEY 。我們隨便看個例子：K 的字鍵代碼爲 203，因此當您敲下K時 110 那一列（ GOSUB KEY ）會變爲相當於 GOSUB 203 。這使得程式跳到203 那一列：

```
203 P = 128: RETURN: REM K
```

這一列將音高(P)設定爲 128 。

之後，程式 RETURN 至 40 那一列：

```
40 POKE 0,P: POKE 1,D:CALL 2
```

這一列將所需的音高（此時爲 128 ）和拍長POKE 到記憶體內，再呼叫機器語言副程式發出該音符。120 那一列可讓您用遊戲控制器控制音符的拍長D：

```
120 D = PDL(0)
```

將控制器旋鈕向左轉到底（ D = 0 ）會使拍長變得很短，因而產生惱人的音質；將它向右轉到底（ D = 255 ）會使時間拖得很長，像風琴的音符一樣，使您

在彈曲時不得不彈得非常慢。您最好由實驗找出您最喜歡的拍長。若想做合理的折衷，可將120 那一列的D設定爲 30：

```
120 D = 30
```

129 至 222 那幾列構成一個對照表（ look-up table ），負責指定音高給每個鍵。本程式將中間一排鍵（ ASDFGHJKL；）指定給最低八度音，P由A鍵的255 開始，只要用您的中指尖由左向右掃過這排鍵，您就會聽到一階音符。再上面一排鍵（ QWERTYUIOP ）比這些鍵各高半音，因此W＝“A#”，E＝“S#”，R＝“D#”，…，以此類推，就好像鋼琴上的黑白鍵一樣。

最上面一排鍵（1234567890：－）高八度音，而底排（ZXCVBNM,./ ）又繼續高八度音。對所有三個八度音，您可以利用 SHIFT和CONTROL 鍵爲相關各鍵提高半音；比方說，對第一排而言，按 3 會得到一個音符，而按 SHIFT－3（按著SHIFT， 然後按 3 ）會將它提昇半音。對於字母鍵， SHIFT 鍵沒有作用，因爲K與 SHIFT－K 的字鍵代碼完全相同（都是203 ），但CONTROL－K呼叫的139 那一列所包含的音高則比由K呼叫的 203 那一列高半音：

```
139 P = 120:RETURN:REM C/K
203 P = 128:RETURN:REM K
```

因此，若要使下面兩排文字鍵（ ASDFGHJKL；或ZXCVBNM,./ ）之中的任何一個鍵昇半音，只要按著CONTROL 再敲該鍵（除了CONTROL-C，它在Applesoft 內的作用將會使程式脫出）。注意，這完全是爲了方便起見，讓您有兩種方法來提昇主要鍵（ ASDFGHJKL；）的音調：按著CONTROL鍵，或向上移一排都可以。最後，最高（第四個）八度音可利用CONTROL和最上面的一排字母鍵（ QWERTYUIOP ）而獲得，但它們沒有高半音鍵。所以，整個說起來，您有四個八度音，下面三個八度音包含所有您在鋼琴上可從黑色鍵獲得的升半音

和降半音音符。

• 若想脫出程式可按空白鍵（字鍵代碼為160 ），它可使程式跳到：

```
160 END
```

Organ

```
5 REM    APPLE ORGAN NEW, WITH SEMITONES -----
8 DIM PP(255): REM  STORE NOTES FOR PLAYBACK
10 GOSUB 32000
11 PRINT : PRINT : PRINT : PRINT
12 TEXT : VTAB 24
14 PRINT "HIT A FEW KEYS & SEE WHAT HAPPENS"
15 PRINT "HIT SPACE BAR TO EXIT"
16 PRINT "HIT SHIFT-1 (!) FOR PLAYBACK"
18 GR
20 KEY= PEEK (-16384)
30 IF KEY>127 THEN GOTO 100
40 POKE 0,P: POKE 1,D: CALL 2
50 GOTO 20
100 POKE -16368,0: REM  CLEAR KEYBOARD
110 GOSUB KEY
112 COLOR= RND (16): PLOT X,P/6-3:X=X+1
114 IF X>39 THEN GR
116 IF X>39 THEN X=0
120 D=30
121 PP(K)=P: REM  SAVE NOTE FOR PLAYBACK
122 K=K+1
125 KEY=0
126 GOTO 20
129 P=242: RETURN : REM   C/A
130 P=40: RETURN : REM  C/B
132 P=192: RETURN : REM   C/D
133 P=25: RETURN : REM  C/E
134 P=182: RETURN : REM  C/F
135 P=161: RETURN : REM  C/G
136 P=142: RETURN : REM  C/H
137 P=16: RETURN : REM  C/I
138 P=128: RETURN : REM   C/J
139 P=120: RETURN : REM   C/K
140 P=108: RETURN : REM   C/L
141 P=32: RETURN : REM   C/M
142 P=36: RETURN : REM   C/N
143 P=14: RETURN : REM   C/O
144 P=12: RETURN : REM   C/P
145 P=32: RETURN : REM    C/Q
146 P=24: RETURN : REM  C/R
147 P=212: RETURN : REM  C/S
148 P=21: RETURN : REM  C/T
```

待續…

```
149 P=17: RETURN : REM  C/U
150 P=45: RETURN : REM  C/V
151 P=28: RETURN : REM  C/W
152 P=53: RETURN : REM  C/X
153 P=19: RETURN : REM  C/Y
154 P=60: RETURN : REM  C/Z
155 P=255: RETURN : REM  ESC
160 END
161 GOSUB 250: REM  PLAYBACK:SHIFT-1(!)
162 P=106: RETURN : REM  "
163 P=96: RETURN : REM  #
164 P=91: RETURN : REM  $
165 P=80: RETURN : REM  %
166 P=73: RETURN : REM  &
167 P=68: RETURN : REM  '
168 P=60: RETURN : REM  (
169 P=53: RETURN : REM  )
170 P=45: RETURN : REM  *
171 P=96: RETURN : REM  +
172 P=32: RETURN : REM  ,
173 P=42: RETURN : REM  -
174 P=28: RETURN : REM  .
175 P=25: RETURN : REM  /
176 P=50: RETURN : REM  0
177 P=128: RETURN : REM  1
178 P=112: RETURN : REM  2
179 P=100: RETURN : REM  3
180 P=96: RETURN : REM  4
181 P=86: RETURN : REM  5
182 P=75: RETURN : REM  6
183 P=67: RETURN : REM  7
184 P=64: RETURN : REM  8
185 P=56: RETURN : REM  9
186 P=48: RETURN : REM  :
187 P=100: RETURN : REM  ;
188 P=30: RETURN : REM  <
189 P=40: RETURN : REM  =
190 P=27: RETURN : REM  >
191 P=24: RETURN : REM  ?
192 P=104: RETURN : REM  @
193 P=255: RETURN : REM  A
194 P=42: RETURN : REM  B
195 P=50: RETURN : REM  C
196 P=202: RETURN : REM  D
197 P=212: RETURN : REM  E
198 P=192: RETURN : REM  F
199 P=172: RETURN : REM  G
200 P=150: RETURN : REM  H
201 P=135: RETURN : REM  I
202 P=135: RETURN : REM  J
203 P=128: RETURN : REM  K
204 P=112: RETURN : REM  L
205 P=34: RETURN : REM  M
206 P=38: RETURN : REM  N
207 P=120: RETURN : REM  O
208 P=108: RETURN : REM  P
209 P=255: RETURN : REM  Q
210 P=202: RETURN : REM  R
211 P=224: RETURN : REM  S
212 P=182: RETURN : REM  T
213 P=142: RETURN : REM  U
214 P=48: RETURN : REM  V
215 P=240: RETURN : REM  W
216 P=56: RETURN : REM  X
217 P=161: RETURN : REM  Y
218 P=64: RETURN : REM  Z
221 P=32: RETURN : REM  M-SHIFT
222 P=37: RETURN : REM  `
250 REM  PLAYBACK -----
254 D=100
258 FOR J=1 TO K
260 P=PP(J)
270 POKE 0,P: POKE 1,D: CALL 2
280 NEXT J
290 K=0
300 RETURN
32000 POKE 2,173: POKE 3,48: POKE 4,192: POKE 5,136: POKE 6,208: POKE 7,4
      : POKE 8,198: POKE 9,1: POKE 10,240
32005 POKE 11,8: POKE 12,202: POKE 13,208: POKE 14,246: POKE 15,166: POKE
      16,0: POKE 17,76: POKE 18,2: POKE 19,0: POKE 20,96: RETURN
```

音樂(Music)

本程式可從Apple 引出音樂，它當然不可能是交響樂的水準。程式由 370 至 390 那幾列開始，先將一些數目 POKE 到 770 至 790 的記憶位址（ memory address ）內，其餘部份則為一連串音符選擇音高P和拍長D。 270 那一列將P和D POKE 到記憶體內，然後 CALL 儲存在記憶位址 770 的機器語言副程式：

```
270 POKE 768, P: POKE 769, D: CALL 770
```

音高和拍長的數值都必須介於 0 與 255 之間。注意，增加D的數值會增加每一個音符的長度，而增加 P 的數值則會降低音高（ 因為音高＝頻率＝1／週期）。P與D的數值可經由遊戲控制器於 200 至 280 那幾列輸入：

```
210 D = PDL (1)
220 P = PDL (0)
```

另外，遊戲者可要求 300 至 360 那幾列彈奏半音階，其初始音高由 310 那一列定於 255 ，然後由 320 那一列一次提昇半音：

```
320 P = P / 1.0595: REM RAISE BY ONE SEMITONE
```

您可能在學校的物理課中讀過將一個音符的頻率加倍可將它的音高提昇八度音，因此下式：

```
P = P / 2
```

可將一個音符的週期(P)減半，而使其頻率加倍，將音高提昇八度音。一個八度音有 12 個半音（ 在鋼琴上算一算琴鍵 ），而 2 的十二次方根為1.0595 ；因此，320 那一列中將週期除以 1.0595 可將音符的頻率增加 5.95%而使音高提昇半音。

從以上說明，我們可以看出如何將曲子放在 DATA 敍述內，令 Apple

READ 它們，並演奏它們。試做下列修改：

```
110 PRINT "DATA TUNES"
290 REM DATA TUNES---
300 D = 30: REM DURATION
310 KEY = 255: REM INITIAL PITCH = KEY OF C
320 READ Q
330 IF Q > 255 THEN RSTORE: REM REPLAY
340 P = KEY / (1.0595 ^ Q): REM NUMBER OF
SEMITONES ABOVE KEYNOTE
350 POKE 768, P: POKE 769, D: CALL 770
360 GOTO 320
....
500 DATA 0, 3, 6, 9, 12, 15, 18
510 DATA 999
```

500 那一列內的 DATA 包含一首一連串以高於主鍵多少半音數來表示音符的曲子。340 那一列將這些編了碼的 DATA 敍述轉換成各種音符的實際音高。

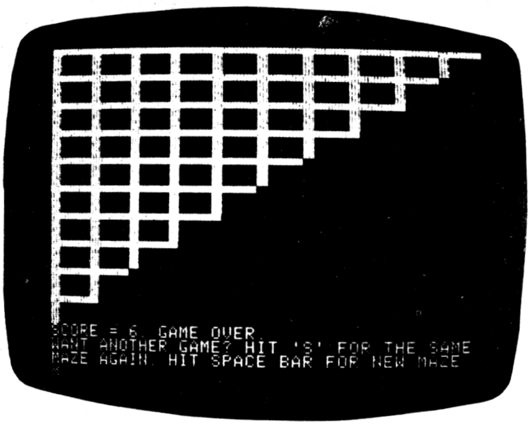
Music

```
10   REM *****************
20   REM *               *
30   REM *   APPLE MUSIC *
40   REM *               *
50   REM *****************
60   REM
70   GOSUB 380: REM  TONES SUBROUTINE ---
80   TEXT : HOME
90   PRINT "APPLE MUSIC": PRINT
100   PRINT "WOULD YOU LIKE:--": PRINT
110   PRINT "1. SCALES"
120   PRINT "2. PADDLE MUSIC"
130   GET N$:N =  VAL (N$)
140   IF N <  > 1 AND N <  > 2 THEN 80
150   ON N GOTO 290,160
160   PRINT "PADDLE (0) SETS THE PITCH"
170   PRINT "PADDLE (1) SETS THE DURATION"
180   PRINT "HIT ANY KEY TO START THE 'MUSIC'"
190   GET H$
200   REM   PADDLE PLAYS A TUNE-----
210 D =  PDL (1)
220 P =  PDL (0)
230  VTAB 10: HTAB 7: PRINT "     ";
240  VTAB 10: HTAB 26: PRINT "     ";
250  VTAB 10: HTAB 1
260  PRINT "PITCH="; PDL (0),"DURATION="; PDL (1)
```

待續

```
270  POKE 768,P: POKE 769,D: CALL 770
280  GOTO 210
290  REM SCALES-----
300 D = 30: REM  DURATION
310 P = 255: REM  INITIAL PITCH
320 P = P / 1.0595: REM  RAISE BY ONE SEMITONE
330  IF P < 25 THEN 80
340  VTAB 22: PRINT "PITCH="; INT (P);" ","DURATION=";D
350  POKE 768,P: POKE 769,D: CALL 770
360  GOTO 320
370  REM  INITIAL POKES-----
380  POKE 770,173: POKE 771,48: POKE 772,192: POKE 773,136: POKE 774,208:
      POKE 775,5: POKE 776,206: POKE 777,1: POKE 778,3: POKE 779,240: POKE
     780,9: POKE 781,202
390  POKE 782,208: POKE 783,245: POKE 784,174: POKE 785,0: POKE 786,3: POKE
     787,76: POKE 788,2: POKE 789,3: POKE 790,96: RETURN
```

迷宮(Elithorn)



用這個迷宮測驗看看自己的智力

本程式可畫出一個不同於一般形式的迷宮，它是英國心理學家Alec Elithorn 發明的，可用來衡量成人的智力，您可以試試看！

您要從迷宮的左上角開始，利用左箭號鍵（←）或右箭號鍵（→）向下移或向右移。因爲不可以向上移或向左移，因此不能後退，必須勇往直前。電腦會隨機挑選一些交叉點作爲目標點，並將它們塗上藍色，您的任務是設計一條路線儘可能地包含這些目標點。

您可以重複玩同一幅迷宮圖，試試各種不同路線。不僅如此，您也可以與朋友們比比IQ ，每個人都玩同一幅迷宮圖，得分最高的自然IQ也最高 ，可以請其他人每人一杯飲料。

200 至 250 那幾列描繪水平線與垂直線互相交叉形成的方格，而 150 那一列向遊戲者要求一個介於 1 與 5 之間的數目N，以決定彩色目標點的多寡。270 至 370 那幾列內的副程式求取GOAL（J，K） 陣列的數值，並據以決定那些目標點

要放在那裡，並在 360 那一列內畫出來。

遊戲的主迴圈在 390 至 490 那幾列內，它根據遊戲者所按的鍵描繪遊戲者的位置。420 和 430 那兩列檢查遊戲者是否按了倒退鍵（左箭號，字鍵代碼爲 136）或前進鍵（右箭號，字鍵代碼爲 149），以便藉增量 X 或 Y 向右或向下移動。程式接著跳到 510 那一列的副程式，並於 530 那一列檢查剛剛選擇的小方塊（SCRN(X, Y)）是否爲彩色目標點；如果它是，580 那一列便將遊戲者的分數加一。遊戲者的新動向是在 460 至 490 那幾列中描繪的，然後 390 至 490 那幾列的主迴圈就一直等到遊戲者進行下一步。

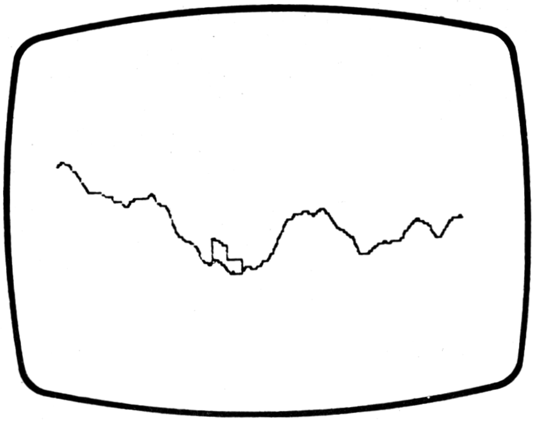
Elithorn

```
10   REM   ******************
20   REM   *     ELITHORN   *
30   REM   *        BY      *
40   REM   * STUART ANSTIS  *
50   REM   ******************
60   REM
70   REM
80   GOSUB 860
90   TEXT : HOME : VTAB 10
100   PRINT "WANT INSTRUCTIONS (Y/N) ?"
110   GET Y$
120   IF Y$ = "Y" THEN  GOSUB 700
130   PRINT : PRINT : PRINT : PRINT
140   PRINT "DO YOU WANT A FEW COLORED POINTS (1) OR A LOT (5) ? TYPE IN
      A NUMBER FROM 1 TO 5"
150   GET N$:N =  VAL (N$): IF N < 1 THEN 150
160   DIM GOAL(40,40)
170  SUCCESS = 0
180   TEXT : HOME : GR
190  BARS = 14:GOAL = 5: REM  COLORS
200   REM  PLOT GRID -----
210   COLOR= BARS
220   FOR I = 0 TO 39 STEP 4
230   HLIN 0,39 - I AT I
240   VLIN 0,39 - I AT I
250   NEXT I
```

待續

```
270  REM  PLOT COLORED GOAL POINTS -----
280 NUMBER = 1 - N / 10
290  COLOR= GOAL
300  FOR J = 0 TO 39 STEP 4
310  FOR K = 0 TO 40 - J STEP 4
320  IF FLAG = 1 THEN  GOTO 360
330 GOAL(J,K) = 0: REM  CLEAR ARRAY -----
340 R =  RND (100)
350  IF R > NUMBER THEN GOAL(J,K) = 1: REM  FILL ARRAY -----
360  IF GOAL(J,K) = 1 THEN  PLOT J,K: REM  PLOT ARRAY -----
370  NEXT K: NEXT J
380 X = 0:Y = 0
390  REM  CHECK WHICH KEY PRESSED -----
400 KEY =  PEEK ( - 16384)
410  IF X + Y > 38 THEN  GOTO 610
420  IF KEY = 136 THEN Y = Y + 4
430  IF KEY = 149 THEN X = X + 4
440  IF KEY > 127 THEN  GOSUB 510
450  GOTO 400
460  REM   PLOT SQUARE -----
470  COLOR= 15
475  IF X > 39 THEN X = 39
480  PLOT X,Y
490  RETURN
500  REM  CHECK IF COLORED GOAL POINT REACHED -----
510  POKE  - 16368,0
520 COUNT =  SCRN( X,Y)
530  IF COUNT = GOAL THEN  GOSUB 560
540  GOSUB 460
550  RETURN
560  REM  SCORE AT COLORED GOAL POINTS -----
570 D = 30:P = 50 + 200 *  RND (1): GOSUB 850
580 SUCCESS = SUCCESS + 1
590  VTAB 23: PRINT " SCORE = ";SUCCESS;"  "
600  RETURN
610  PRINT "SCORE = ";SUCCESS;". GAME OVER.  "
620  FOR J = 1 TO SUCCESS
630 P = 200 / 1.1225 ^ J:D = 20
640  GOSUB 850
650  NEXT J
660  PRINT "WANT ANOTHER GAME? HIT 'S' FOR THE SAME MAZE AGAIN. HIT SPACE
     BAR FOR NEW MAZE"
670  GET X$
680  IF X$ = "S" THEN FLAG = 1: GOTO 170
690 FLAG = 0: GOTO 170
700  REM  INSTRUCTIONS ------
710  TEXT : HOME : VTAB 10
720  PRINT "THIS MAZE WAS DEVISED BY THE BRITISH"
725  PRINT "PSYCHOLOGIST ALEC ELITHORN."
730  PRINT "YOU START AT TOP LEFT CORNER.  HIT BACK-ARROW TO MOVE DOWN; H
     IT FORWARD-ARROW TO MOVE RIGHT.  YOU CANNOT RETRACE YOUR"
735  PRINT "PATH."
740  PRINT "YOU SCORE ONE POINT FOR EACH COLORED"
745  PRINT "INTERSECTION YOU GO THROUGH."
750  PRINT "AFTER RUNNING THE MAZE, YOU CAN HIT"
755  PRINT "'S' TO TRY THE SAME MAZE AGAIN."
760  PRINT
770  PRINT "   ***   GOOD LUCK !! ***"
780  PRINT : PRINT : PRINT
790  INVERSE
800  PRINT "  HIT SPACE BAR TO CONTINUE   "
810  NORMAL
820  GET K$
830 FLAG = (K$ = "S")
840  RETURN
850  POKE 768,P: POKE 769,D: CALL 770: RETURN
860  POKE 770,173: POKE 771,48: POKE 772,192: POKE 773,136: POKE 774,208:
      POKE 775,5: POKE 776,206: POKE 777,1: POKE 778,3: POKE 779,240: POKE
     780,9: POKE 781,202
870  POKE 782,208: POKE 783,245: POKE 784,174: POKE 785,0: POKE 786,3: POKE
     787,76: POKE 788,2: POKE 789,3: POKE 790,96: RETURN
```

爬坡(Running on Empty)



這條崎嶇不平的路，可測驗您省油的天賦

這是一個高解析度的單人遊戲，在遊戲裡您駕著一部車，帶著一定數量的燃料沿著一條非常崎嶇不平的路開。遊戲的目的不是追求速度，而是講究經濟，您要在用盡燃料之前儘可能地跑得愈遠愈好。當車子跑到螢光幕的右邊時它會繞回左邊重新出現，每橫過螢光幕一次就算跑了一圈。您要利用遊戲控制器作爲油門，您的數値（介於 0 與 255 之間）會出現在螢光幕的左下方，您轉得愈高，燃料就燒得愈快。另一方面，如果您轉得太慢，車子很可能爬不過某些坡度，一直在原地掙扎和浪費燃料。您可以愛開多少次就開多少次，試試各種不同方法來改善您的操作；比方說，懶人的方法就是把油門定在某個定速，然後每次只改變油門的設定値看看其效果（其結果可能會令您很驚訝！）。希望您能因而找到控制油門的訣竅，讓您的車子以最省油的方式剛剛可爬過每一座所遇的山坡。

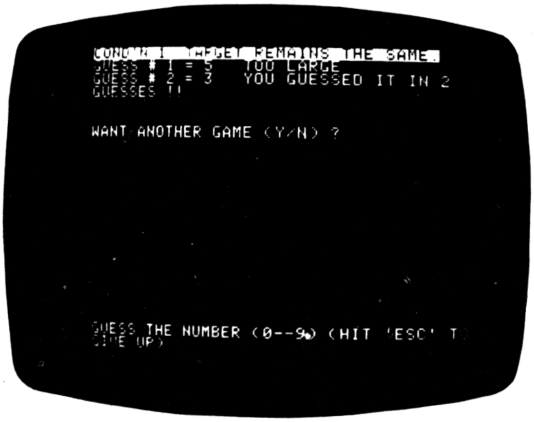
Running On Empty

```
10   REM ************************
20   REM *                      *
30   REM *   RUNNING ON EMPTY   *
40   REM *                      *
50   REM *   BY STUART ANSTIS   *
60   REM *                      *
70   REM ************************
80   REM
90   REM  INITIALISE -----
100   TEXT : HOME
110  BUMP = 1: REM  SIZE OF BUMPS ON ROAD
120  G = 15: REM    GRAVITY
130  F = 1000: REM    FUEL
140   DIM Y(280)
150  Y(0) = 80
160   HGR : HCOLOR= 7
170   HPLOT 0,Y(0)
180   REM  PLOT BUMPY ROAD -----
190   FOR X = 1 TO 279
200  DY = BUMP *  RND (1) - BUMP *  RND (1) + 0.8 * DY
210  Y(X) = Y(X - 1) + DY
220   HPLOT  TO X,Y(X)
230   NEXT X
240   VTAB 21: PRINT "THROTTLE","FUEL","LAPS";
250   REM  MAIN LOOP -----
260  X = 1
270  P0 =  PDL (0): REM  THROTTLE
280  F =  INT (F - P0 * P0 / 10000): REM  USE UP FUEL
290  GG = G * (Y(X) - Y(X - 1)): REM  SLOPE OF HILL
300  INCR = (4 * P0 + GG * GG *  SGN (GG)) / 200
310   VTAB 22: HTAB 1: PRINT "     ","     ","      ";
320   VTAB 22: HTAB 1: PRINT P0,F,LAPS; INT (100 * X / 250) / 100
330   IF F < 1 THEN  GOTO 440
340  X = X + INCR
350   IF X > 250 THEN LAPS = LAPS + 1:X = 1
360   IF X < 1 THEN X = 1
370   REM  PLOT CAR -----
380   FOR HC = 7 TO 0 STEP P - 7
390   HCOLOR= HC
400   HPLOT X,Y(X) - 1 TO X,Y(X) - 10 TO X + 10,Y(X + 10) - 10 TO X + 10,Y
      (X + 10) - 5 TO X + 20,Y(X + 20) - 5 TO X + 20,Y(X + 20) - 1
410  K =  PEEK ( - 16336) +  PEEK ( - 16336)
420   NEXT HC
430   GOTO 270
```

待續...

```
440   REM   OUT OF FUEL -----
450   CALL  - 1059: CALL  - 1059
460   VTAB 23: HTAB 1: PRINT "    *****      OUT OF FUEL      *****";
470   VTAB 24: HTAB 1: PRINT "WANT ANOTHER RUN (Y/N)  ";
480   GET Y$
490   HOME
500   IF Y$ = "Y" THEN 590
510   VTAB 24: HTAB 1: PRINT "WANT A FRESH ROUTE (Y/N)";
520   CALL  - 1059
530   GET R$
540   IF R$ = "Y" THEN  HOME : GOTO 590
542   PRINT "WANT TO RETURN TO MENU (Y/N) ?"
544   GET M$
546   IF M$ = "Y" THEN  TEXT : PRINT : PRINT  CHR$ (4);"RUN HELLO"
550   VTAB 21: PRINT "OK.  THANKS FOR PLAYING."
560   FOR T = 1 TO 1000: NEXT T
570   TEXT : HOME
580   END
590  F = 1000: REM  FUEL
600  LAPS = 0
610   IF R$ = "Y" THEN  GOTO 150
620   GOTO 240
```

數字猜謎(Guess a Number)



教您的Apple玩玩這個古老遊戲

這個遊戲有容易的版本，也有困難的版本，當您 RUN 本程式時這兩種都會存在。在容易的版本裡，電腦所提供的數目介於 0 與 9 之間，您要設法猜出該數目，電腦會在您猜測時提示您TOO LARGE(太大)或TOO SMALL(太小)或YES, YOU GUESSED IT!(是的，您猜對了！)。

您要猜對一個數目所需要的次數最少是幾次呢？明顯地，您當然可以用 0? 1? 2? 3? 4? 5? 6? 7? 8? 9? 方式一個一個地猜，運氣最不好的時候(大約十次中有一次)也只需要問十次而已；也就是電腦提出的數目剛好是您最後一個問到的數目。用這種方法玩，長時間來說，每題大約需要答五次。但是，到底有沒有更好的法子來猜亂數呢？有一個方法已經被證明很有效，尤其當亂數的範圍很大的時候。這種方法若用來猜一個介於 1 與 128 之間的數目平均起來大約需要 7 次，但對於一個介於 1 與 1024 之間的數目也只不過大約需要 10 次而已；因此，當數目的範圍增加約 10 倍時所需猜測的數目才增加 3 次。這種方法是一種"連

續中分法”，其進行狀況如下：

```
GUESS A NUMBER BETWEEN 1 AND 128
64            ....#1
TOO LARGE
32            ....#2
TOO LARGE
16            ....#3
TOO SMALL
24            ....#4
TOO LARGE
20            ....#5
TOO SMALL
22            ....#6
TOO SMALL
23            ....#7
YOU GUESSED IT!
```

爲什麼這種方法如此有效呢？（想想看！）您可以當做一個練習，寫一個程式讓電腦利用這種方法自己去猜其所提出的數目。

本遊戲可以做得更難而且更有趣一些，只要稍微改一下：讓N在您每猜一次之後就加1。假設N由2開始。

```
RUN                     (N = 2)
TYPE IN GUESS #1...5
TOO LARGE               (N INCREASES TO 3)
TYPE IN GUESS #2...4
TOO LARGE               (N INCREASES TO 4)
TYPE IN GUESS #3...3
TOO SMALL
```

在標準遊戲中，N的數值是保持固定的。但上述版本更可以訓練一個人的邏輯思考能力。此外，當 N=9 而又要增加1時，它會折回0：

```
RUN                     (N = 9)
TYPE IN GUESS #1...7
TOO SMALL               (N = 9 +1, BUT WRAPS
AROUND TO 0)
TYPE IN GUESS #2...7
TOO LARGE
```

這種較難玩的版本還有一層花樣：在遊戲當中，N在每一次猜測之後是加一或減一遊戲者是不會知道的！

Guess A Number

```
10  REM ********************
20  REM *                  *
30  REM *  GUESS A NUMBER  *
40  REM *                  *
50  REM *        BY        *
60  REM *                  *
70  REM *  STUART ANSTIS   *
80  REM *                  *
90  REM ********************
100  REM
110  DIM N(40),G(40)
120  TEXT : HOME : VTAB 10
130  PRINT "WANT INSTRUCTIONS (Y/N) ? "
140  GET Y$
150  IF Y$ = "Y" THEN  GOSUB 790: REM   INSTRUCTIONS
160  FOR J = 1 TO 30:N(J) = 0:G(J) = 0: NEXT J
170  HOME : VTAB 5
180  PRINT : PRINT "DO YOU WANT:--"
190  PRINT
200  PRINT "1. EASY: TARGET NUMBER DOES NOT CHANGE"
210  PRINT
220  PRINT "2. DIFFICULT: TARGET NUMBER INCREASES BY +1 AFTER EACH GUESS"

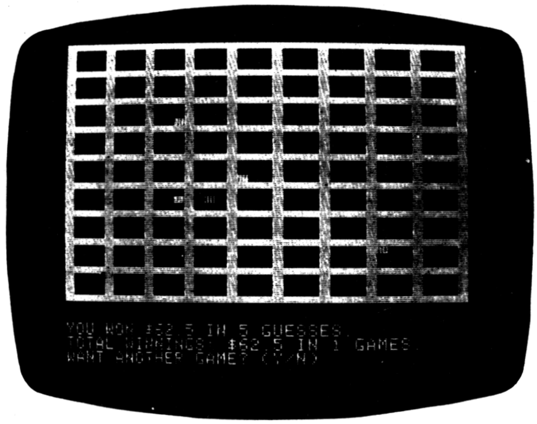
230  PRINT
240  PRINT "3. VERY DIFFICULT: TARGET NUMBER EITHER INCREASES BY +1 OR DE
     CREASES BY -1 AFTER EACH GUESS, ";
250  PRINT "BUT YOU WON'T KNOW WHICH."
260  PRINT
270  PRINT : PRINT "HIT 1, 2 OR 3"
280  GET H$:H =  VAL (H$)
290  HOME
300  IF H < 1 OR H > 3 THEN 180
310  IF H = 1 THEN K = 0
320  IF H = 2 THEN K = 1
330 R =  RND (8)
340  IF H = 3 THEN K = (R > 0.5) - (R < 0.5): REM  +1 OR -1
350 N(1) =  INT ( RND (3) * 10)
360  GOSUB 710
370  REM  MAIN LOOP -----
380  FOR J = 1 TO 40
390  VTAB 21
400  FLASH
410  PRINT "GUESS THE NUMBER (0--9) (HIT 'ESC' TO": PRINT "GIVE UP)"
420  NORMAL
430  GET G$
440  IF G$ =  CHR$ (27) THEN 1170: REM  GIVE UP
450  IF J = 40 THEN 1170: REM  FAILED AFTER 40 GUESSES
460 G(J) =  VAL (G$)
470 JJ = J
480  IF J > 20 THEN JJ = J - 20
490  IF J = 21 THEN  HOME : GOSUB 710
500  VTAB JJ + 1
510  PRINT "GUESS # ";J;" = ";G(J),
520  IF G(J) < N(J) THEN  PRINT "TOO SMALL"
530  IF G(J) > N(J) THEN  PRINT "TOO LARGE"
540  IF G(J) = N(J) THEN  PRINT "YOU GUESSED IT IN ";J: PRINT "GUESSES !!
     ": PRINT : PRINT : GOTO 610
550 JMAX = J
560 NN = N(J) + K
570 N(J + 1) = 10 * (NN / 10 -  INT (NN / 10)): REM   NN MOD 10
580 N(J + 1) =  INT (N(J + 1) + 0.5)
590  NEXT J
600  PRINT
610  PRINT "WANT ANOTHER GAME (Y/N) ?";
620  GET Y$
630  IF Y$ = "Y" THEN 160
640  HOME : VTAB 10
650  PRINT "HIT 'ESC' TO EXIT"
660  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
670  GET K$
680  IF K$ =  CHR$ (27) THEN  HOME : PRINT "OK. THANKS FOR PLAYING": END

690  PRINT : PRINT  CHR$ (4);"RUN HELLO"
700  END
```

```
710  REM  PRINT CONDITION -----
720  INVERSE
730  PRINT "COND'N ";H;": TARGET ";
740  IF H = 1 THEN  PRINT "REMAINS THE SAME."
750  IF H > 1 THEN  PRINT "INCREASES"
760  IF H = 3 THEN  VTAB 1: HTAB 27: PRINT " OR DECREASES"
770  NORMAL
780  RETURN
790  REM  INSTRUCTIONS -----
800  HOME
810  HTAB 15: PRINT "GUESS A NUMBER
820  PRINT
830  FOR J = 1 TO 8: PRINT "*****";: NEXT J
840  PRINT
850  PRINT "THE COMPUTER WILL THINK OF A NUMBER"
860  PRINT "BETWEEN 0 AND 9 INCLUSIVE.  YOU HAVE TO GUESS THE NUMBER.  ";

870  PRINT "THE COMPUTER WILL ": PRINT "TELL YOU IF YOUR GUESS IS TOO LAR
     GE OR"
880  PRINT "TOO SMALL."
890  PRINT : PRINT "TO MAKE THE GAME HARDER (AND MORE FUN), YOU CAN CHOOS
     E A HARDER VERSION IN WHICH";
900  PRINT "THE TARGET NUMBER INCREASES BY +1 AFTER EACH GUESS."
910  PRINT
920  PRINT "SAMPLE RUN:--"
930  PRINT
940  PRINT "TARGET NO.=7. GUESS#1=8. TOO LARGE"
950  PRINT "TARGET NO.=8. GUESS#2=7. TOO SMALL"
960  PRINT "TARGET NO.=9. GUESS#3=9. YOU GUESSED IT IN 3 GUESSES !"
970  PRINT
980  INVERSE
990  PRINT "HIT ANY KEY TO CONTINUE...";
1000  NORMAL
1010  GET HIT$
1020  HOME : VTAB 5
1030  PRINT "IF THE TARGET IS 9, THEN ADDING +1 MAKES IT JUMP BACK TO 0 A
     GAIN."
1040  PRINT : PRINT "FOR AN EVEN MORE DIFFICULT GAME, THE"
1045  PRINT "COMPUTER WILL EITHER ADD 1 OR SUBTRACT 1";
1050  PRINT "FROM THE TARGET AFTER EACH GUESS, BUT"
1060  PRINT "YOU WON'T KNOW WHICH!"
1070  INVERSE
1075  PRINT : PRINT "HIT ANY KEY TO CHOOSE YOUR GAME..."
1078  NORMAL
1080  GET Y$
1090  RETURN
1170  REM  GIVE UP---
1180  PRINT "THE CURRENT TARGET NUMBER WAS ";K(J)
1190  GOTO 610
```

尋寶(Buried Treasure)



格子角上的小方塊會暗示您下一步到那裡去尋寶

在本遊戲中，有一包 $1000 元的獎金藏在方格圖上某個地方，您要在10 那一列選擇方格圖的大小，並旋轉1 號遊戲控制器移動一個彩色小標記到一個方格中藉按下控制器按鈕去挖掘寶藏。如果您挖到該寶藏，您就可以擁有那些錢，如果沒找著，則會有一個藍色小方塊發亮，粗略指向藏寶的方向；比方說，如果藍色小方塊在標記的東北方，您就知道該寶藏在東北方的某個方格中。（這種手法很接近第一章中說明的飛彈導引方法。）然而，您每猜錯一次寶藏就會減爲一半；如果您第二次才找到它，則您只能得到 $500 元，第三次才找到只能得 $250元，…，以此類推。您可以愛玩多少次就玩多少次，電腦會告訴您每次贏得多少錢，以及累積贏得的總數。

您可能已注意到，本遊戲正如〝呑食蛋糕〞（Chomp）遊戲一樣，是Nim遊戲的一種二維空間版本，因此〝尋寶〞遊戲實際上是一種二維空間的數字猜謎遊戲。〝尋寶〞遊戲是利用具體的低解析度平面圖形，而數字猜謎則是概念與抽象

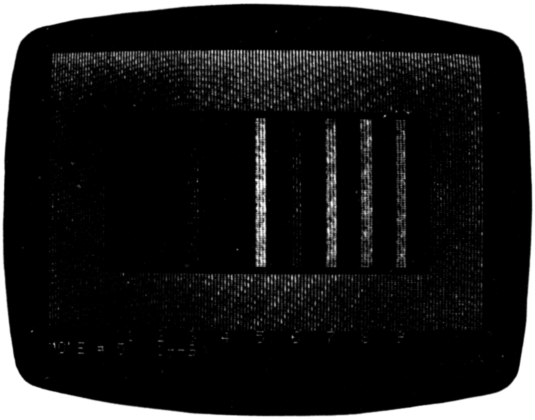
的，但基本的邏輯觀念以及使用的策略則是相同的。

Buried Treasure

```
10  REM   BURIED TREASURE-----
20  TEXT : HOME : VTAB 3
30  PRINT "BURIED TREASURE: MAKE SURE PADDLES ARE"
40  PRINT "PLUGGED IN !": PRINT
50  PRINT "WANT INSTRUCTIONS? (Y/N) "
60  GET Y$
70  IF Y$ = "Y" THEN  GOSUB 510
80  PRINT "HOW MANY SQUARES PER SIDE (3-9)": GET SIZE
90  REM A,B=COORDINATES OF TREASURE---
100 A =  INT (SIZE *  RND (1)):B =  INT (SIZE *  RND (2))
110 GUESS = 0:GOLD = 2000
120  REM DRAW GRID-----
130  TEXT : HOME : GR
140  COLOR= 15
150  FOR I = 0 TO 4 * SIZE STEP 4
160  HLIN 0,4 * SIZE AT I
170  VLIN 0,4 * SIZE AT I
180  NEXT I
190  REM  PADDLE # 1 DETERMINES POSITION OF SQUARE-----
200 PADDLE =  PDL (1) * SIZE * SIZE / 256
210 POLD =  PDL (1)
220 XX =  INT (SIZE * (PADDLE / SIZE -  INT (PADDLE / SIZE))): REM PADDLE
     MOD SIZE
230 YY =  INT (PADDLE / SIZE)
240 X = 4 * XX + 2
250 Y = 4 * YY + 2
260  COLOR= 9
270  PLOT X,Y
280  IF  ABS (POLD -  PDL (1)) > 1 THEN  GOTO 320
290  IF  PEEK ( - 16286) > 127 THEN  GOTO 330
300  GOTO 280
310  REM  ERASE OLD POSITION-----
320  COLOR= 0: PLOT X,Y: GOTO 200
330  REM  CHECK PUSHBUTTON GUESS-----
340 GOLD = GOLD / 2
350 GUESS = GUESS + 1
```

```
360  REM PLOT H,V=POINTER TO TREASURE---
370 H = X +  SGN (A - XX)
380 V = Y +  SGN (B - YY)
390  IF H = X AND V = Y THEN  GOTO 430
400  COLOR= 7: PLOT H,V
410  IF  PEEK ( - 16286) < 127 THEN  GOTO 280
420  GOTO 410
430 GAME = GAME + 1
440 WIN = WIN + GOLD
450  PRINT "YOU WON $";GOLD;" IN ";GUESS;" GUESSES."
460  PRINT "TOTAL WINNINGS: $";WIN;" IN ";GAME;" GAMES."
470  PRINT "WANT ANOTHER GAME? (Y/N) "
480  GET Y$
490  IF Y$ <  > "Y" THEN  TEXT : HOME : PRINT "BYE.": END
500  GOTO 100
510  REM INSTRUCTIONS-----
520  PRINT : PRINT
530  PRINT "YOU GET TO CHOOSE A GRID SIZE. A $1000"
540  PRINT "TREASURE IS BURIED SOMEWHERE IN THE GRID"
550  PRINT "TURN PADDLE #1 TO MOVE A POINTER ALONG"
560  PRINT "THE ROWS (LIKE READING A BOOK). TO DIG"
570  PRINT "A HOLE, PRESS PUSHBUTTON.  IF YOU'RE"
580  PRINT "WRONG, A HELPFUL POINTER POINTS IN THE"
590  PRINT "DIRECTION OF THE TREASURE. EACH GUESS"
600  PRINT "HALVES THE VALUE OF THE TREASURE--$1000"
610  PRINT "IF YOU GET IT FIRST SHOT, $500 IF IT"
620  PRINT "TAKES TWO GUESSES, $250 FOR 3 GUESSES"
630  PRINT : PRINT : PRINT "   ***  GOOD LUCK  !!! ***"
640  PRINT : PRINT "HIT ANY KEY TO START...."
650  GET G$: HOME : RETURN
```

換位遊戲(Jumpers)



這是一個簡單的換位遊戲，螢光幕會顯示您的選擇

這是一個單人猜謎遊戲，要在一塊具有9條垂直條紋的板上玩：左邊四條爲藍色條紋，右邊四條爲黃色條紋，中間是一條空白條紋。您的目標是讓藍，黃條紋互換位置，您每走一步只能將一條彩色條紋移入臨近的空白條紋，或跳越一條彩色條紋進入空白條紋。比方說，剛開始時，第5條爲空白，您可以有4種選擇：第4條（或第6條）可直接移入第5條，或者第3條（或第七條）可以跳過第4條（或第6條）進入第5條；因此，您的第一步可以按3、4、6或7。本程式會讓您知道您到底走了幾步才換完那些條紋，您應該多多練習，讓次數設法減低。

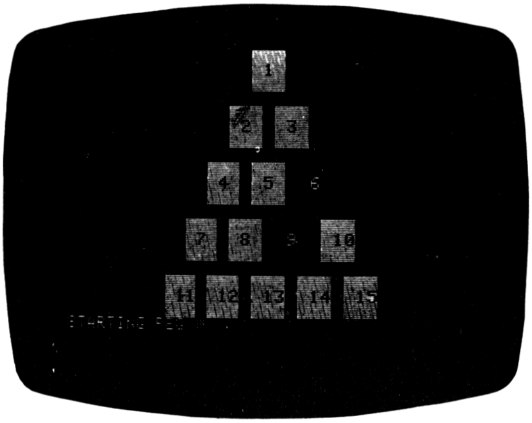
Jumpers

```
10  REM ********************
20  REM *                  *
30  REM *      JUMPERS     *
40  REM *                  *
50  REM *        BY        *
60  REM *                  *
70  REM *  STUART ANSTIS   *
80  REM *                  *
90  REM ********************
100  REM
110  REM
800  TEXT : HOME : VTAB 10
810  PRINT "WANT INSTRUCTIONS (Y/N) ? "
820  GET Y$
830  IF Y$ = "Y" THEN  GOSUB 1600
860  HOME
870  GOSUB 1500: REM  PRINT NUMBERS
880 T = 1: REM  NUMBER OF TURNS
890  GR
900  COLOR= 9
910  FOR I = 0 TO 39
920  HLIN 0,39 AT I
930  NEXT I
940  COLOR= 0
950  FOR I = 9 TO 31
960  HLIN 6,34 AT I
970  NEXT I
1000  FOR X = 1 TO 9
1010 C = 2 * (X < 4.5) + 13 * (X > 5.5)
1020  COLOR= C
1030  VLIN 10,30 AT 3 * X + 5
1040  NEXT X
1100  REM  COLLECT MOVE-----
1110  VTAB 22: HTAB 1
1120  PRINT "MOVE # ";T;"  (1--9) ";
1130  GET M$:M =  VAL (M$): PRINT M;
1140  IF M < 1 OR M > 9 THEN  PRINT : PRINT "SORRY, MUST BE BETWEEN 1 AND
      9     ": GOTO 1120
1200  REM   CHECK LEGALITY OF MOVE-----
1210  IF  SCRN( 3 * M + 5,20) = 0 THEN 1300
1220  FOR J =  - 6 TO 6 STEP 3
1230  IF J = 0 THEN J = 3
1240  IF  SCRN( 3 * M + 5 + J,20) = 0 THEN JJ = J: HOME : GOSUB 1500: GOTO
     1400
1250  NEXT J
1300  PRINT : PRINT "   SORRY, ILLEGAL MOVE--TRY AGAIN"
1310  GOTO 1100
```

```
1400  REM  PLOT THE MOVE-----
1405 S =  SCRN( 3 * M + 5,20)
1410 T = T + 1
1420  COLOR= 0
1430  VLIN 10,30 AT 3 * M + 5
1440  COLOR= S
1450  VLIN 10,30 AT 3 * M + 5 + JJ
1455  GOSUB 1800: REM  CHECK IF GAME OVER
1460  GOTO 1100
1500  REM  PRINT NUMBERS-----
1510  FOR K = 1 TO 9
1520  VTAB 21: HTAB 3 * K + 6: PRINT K;
1530  NEXT K
1540  RETURN
1600  REM  INSTRUCTIONS-----
1610  HOME : VTAB 5
1620  PRINT "IN THIS GAME YOU HAVE 8 BARS ON A 9"
1630  PRINT "HOLE BOARD.  THE OBJECT IS TO"
1640  PRINT "REVERSE THE ORDER OF THE BARS,"
1650  PRINT "AND GET THE DARK BLUE BARS ON THE"
1660  PRINT "LEFT TO REPLACE THE LIGHT YELLOW"
1670  PRINT "BARS ON THE RIGHT, AND VICE VERSA."
1680  PRINT
1690  PRINT "A BAR CAN MOVE INTO AN ADJACENT EMPTY"
1700  PRINT "SPACE, OR JUMP OVER ONE NEIGHBOUR INTO"
1710  PRINT "AN EMPTY SPACE."
1712  PRINT : PRINT "HIT THE NUMBER OF THE BAR YOU WANT TO"
1714  PRINT "MOVE. DO NOT HIT RETURN."
1720  PRINT : PRINT : PRINT
1730  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE  ": NORMAL
1740  GET H$
1750  RETURN
1800  REM  CHECK IF GAME OVER-----
1810  FOR X = 1 TO 9
1820  IF X = 5 THEN X = 6
1830  IF  SCRN( 3 * X + 5,20) <  > 13 * (X < 4.5) + 2 * (X > 5.5) THEN  RETURN

1840  NEXT X
1850  CALL  - 1059: CALL  - 1059
1860  HOME : VTAB 21
1870  PRINT "YOU WON THE GAME IN ";T;" MOVES !!"
1875 T = 1
1880  PRINT "WANT ANOTHER GAME (Y/N) ?"
1890  GET Y$
1900  IF Y$ = "Y" THEN  GOTO 860
1910  TEXT : HOME : VTAB 10
1920  PRINT "HIT 'ESC' TO EXIT"
1930  PRINT : PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
1940  GET H$
1950  IF H$ =  CHR$ (27) THEN  HOME : VTAB 5: PRINT "OK.  THANKS FOR PLAY
    ING.": END
1960  PRINT : PRINT  CHR$ (4);"RUN HELLO"
```

拔釘子(Peggy)



這是另外一個單人猜謎遊戲

這是另外一個單人猜謎遊戲，但是是二維空間的，不像“換位遊戲”（Jumpers）是一維的。“拔釘子”要在一塊帶有15個釘子的三角形板上玩。您可以從任何一支釘子開始拔，之後，您的下一步只能跳越一支釘子到一個空洞去，然後拔去剛剛跳過的那支釘子；您的目標是要使最後留下的釘子愈少愈好，在板上留下4支釘子是很容易達成的，要留下3支釘子則稍微難一些；如果留下2支釘子，則顯示您已經是高手了；1支釘子幾乎是不可能的……但是是可以辦到的！

打入您要起步的釘子號碼和您要跳入之空白號碼，電腦會檢查您的每一步，看看是否符合規則，然後爲您拔去釘子。

Peggy

```
10   REM *******************
20   REM *                 *
30   REM *      PEGGY      *
40   REM *                 *
50   REM *       BY        *
60   REM *                 *
70   REM *  STUART ANSTIS  *
80   REM *                 *
90   REM *******************
100  REM
110  REM
120  TEXT : HOME : VTAB 10
130  PRINT "WANT INSTRUCTIONS (Y/N) ?"
140  GET Y$
150  IF Y$ = "Y" THEN  GOSUB 970: REM  INSTRUCTIONS
160  DIM N(20),V(20),W(20),X(20),H(20),F(20)
170  FOR J = 1 TO 15: READ V(J): NEXT J
180  FOR J = 1 TO 15: READ W(J): NEXT J
190  FOR J = 1 TO 15: READ X(J): NEXT J
200  FOR J = 1 TO 15: READ H(J): NEXT J
210  FOR J = 1 TO 15:N(J) = J: NEXT J
220 NN = 0
230  TEXT : HOME
240  REM  DRAW THE BOARD-----
250 F = 1: REM  INSERT PEGS
260  FOR J = 1 TO 15
270 Q = J
280  GOSUB 1250
290  NEXT J
300  REM  REMOVE INITIAL PEG-----
310  VTAB 21: HTAB 1: INPUT "WHICH PEG WILL YOU REMOVE TO START GAME";S
320  IF S < 1 OR S > 15 THEN  GOSUB 1120: GOTO 310: REM  ILLEGAL MOVE
330  GOSUB 1180: REM  CLEAR BOTTOM LINES
340 F = 0
350 Q = S: GOSUB 1250: REM    REMOVE PEG
360  REM  INPUT LOOP-----
370  VTAB 21: HTAB 1: INPUT "STARTING PEG ";K
380  IF K = 0 THEN 740: REM  GAME OVER
390  INPUT "DESTINATION PEG ";L
400  IF L = 0 THEN 740: REM  GAME OVER
410  IF K > 15 THEN  CALL  - 1059: GOTO 370
420  IF L > 15 THEN  CALL  - 1059: GOTO 370
430  GOSUB 1180: REM  CLEAR LINES
440  REM  CHECK LEGALITY OF MOVE-----
450 V =  ABS (V(K) - V(L)):W =  ABS (W(K) - W(L)):X =  ABS (X(K) - X(L))
460 T(1) = (V = 0 AND W = 2 AND X = 2)
470 T(2) = (V = 2 AND W = 0 AND X = 2)
480 T(3) = (V = 2 AND W = 2 AND X = 0)
490  IF T(1) + T(2) + T(3) = 1 THEN 560: REM  SEPARATION OK
```

```
500  GOSUB 1120: REM  ILLEGAL MOVE
510  GOTO 370
520  CALL  - 1059
530  FOR T = 0 TO 1000: NEXT T
540  GOSUB 1180
550  GOTO 370
560  REM  CHECK IF HOLES FILLED-----
570  IF N(K) <  > 0 AND N(L) = 0 THEN 600
580  GOSUB 1120: REM  ILLEGAL MOVE
590  GOTO 370
600 F = 0:Q = K:N(K) = 0
610  GOSUB 1250: REM  REMOVE PEG K
620 F = 1:Q = L:N(L) = L
630  GOSUB 1250: REM  INSERT PEG L
640  REM  REMOVE JUMPED-OVER PEG-----
650 VV = (V(K) + V(L)) / 2
660 WW = (W(K) + W(L)) / 2
670 XX = (X(K) + X(L)) / 2
680 HH = (H(K) + H(L)) / 2
690  FOR J = 1 TO 15
700  IF V(J) = VV AND W(J) = WW AND X(J) = XX THEN Q = J: GOTO 720
710  NEXT J
720 F = 0:N(Q) = 0:H(Q) = HH: GOSUB 1250: REM  REMOVE PEG
730  GOTO 370
740  REM  GAME OVER-----
750 NN = 0
760  HOME : VTAB 10
770  FOR J = 1 TO 15
780  IF N(J) <  > 0 THEN NN = NN + 1
790  NEXT J
800  PRINT "YOU WON THE GAME WITH ";NN;" PEGS LEFT"
810  PRINT : PRINT "YOUR SCORE WAS ";
820  IF NN > 4 THEN NN = 4
830  ON NN GOTO 840,850,860,870
840  PRINT "FANTASTICALLY GOOD !!": GOTO 880
850  PRINT "VERY GOOD !!": GOTO 880
860  PRINT "QUITE GOOD !!": GOTO 880
870  PRINT "FAIR": GOTO 880
880  PRINT : PRINT "WANT ANOTHER GAME (Y/N) ?"
890  GET Y$
900  IF Y$ = "Y" THEN  GOTO 210
910  HOME : VTAB 8
920  PRINT "HIT 'ESC' TO EXIT"
930  PRINT : PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
940  GET H$
950  IF H$ =  CHR$ (27) THEN  HOME : VTAB 5: PRINT "OK. THANKS FOR PLAYIN
     G": END
960  PRINT : PRINT  CHR$ (4);"RUN HELLO"
970  REM  INSTRUCTIONS-----
980  HOME : VTAB 5
990  PRINT "PEGGY IS A PUZZLE FOR ONE PLAYER,"
1000  PRINT "PLAYED WITH 14 PEGS ON A 15-PEG BOARD"
1010  PRINT "REMOVE ANY ONE PEG TO START. JUMP OVER"
1020  PRINT "AN ADJACENT PEG TO AN EMPTY HOLE."
1030  PRINT "REMOVE JUMPED-OVER PEGS. HIT 0 (ZERO)"
1040  PRINT "WHEN YOU CANNOT MAKE ANY MORE"
1050  PRINT "MOVES. OBJECT IS TO ";
1060  PRINT "END UP WITH AS FEW"
1070  PRINT "PEGS ON THE BOARD AS POSSIBLE."
1080  PRINT : PRINT
1090  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE  "
1100  GET H$
1110  RETURN
1120  REM  ILLEGAL MOVE-----
1130  CALL  - 1059
1140  VTAB 23: HTAB 1: PRINT "SORRY, ILLEGAL MOVE--TRY AGAIN."
1150  FOR T = 1 TO 500: NEXT T
1160  GOSUB 1180: REM  CLEAR LINES
1170  RETURN
1180  REM  CLEAR BOTTOM LINES
1190  HTAB 1
1200  FOR B = 21 TO 23
1210  VTAB B
1220  FOR M = 1 TO 8: PRINT "     ";: NEXT M
1230  NEXT B
1240  RETURN
```

待續 ……

```
1250  REM  REMOVE PEG IF F=0-----
1260  REM  INSERT PEG IF F=1-----
1270 N(Q) = 0
1280  IF F = 1 THEN N(Q) = Q: INVERSE
1290  HTAB H(Q) - 1: VTAB 4 * V(Q) - 2: PRINT "   ";
1300  HTAB H(Q) - 1: VTAB 4 * V(Q) - 1: PRINT " ";Q;: IF Q < 10 THEN  PRINT
 " ";
1310  HTAB H(Q) - 1: VTAB 4 * V(Q): PRINT "   ";
1320  NORMAL
1330  RETURN
1340  REM  V(J)-----
1350  DATA   1,2,2,3,3,3,4,4,4,4,5              ,5,5,5  ,5
1360  REM  W(J)-----
1370  DATA  5,4,5,3,4,5,2,3,4,5,1,2,3,4,5
1380  REM  X(J)-----
1390  DATA  5,5,4,5,4,3,5,4,3,2,5,4,3,2,1
1400  REM  H(J)-----
1410  DATA  20,18,22,16,20,24,14,18,22,26,12,16,20,24,28
```

智慧盤(Fifteen)

這是 Sam Loyd's 很有名的益智遊戲，要在一面有十六塊四方格，上有十五塊小磁磚的板上玩。這些磁磚開始時要打散，您的任務就是一次滑動一塊磁磚進入空白的四方格，一直到您依照下面順序重組好那些磁磚。

```
 1  2  3  4
 5  6  7  8
 9 10 11 12
13 14 15  -
```

磁磚可以上、下、左或右移動，但是不可以做對角線滑動，您只要打入您想移動的磁磚號碼，電腦就會替您移動它。例如，假設一開始磁磚的排列如下：

```
11  4 12 15
 2  9 13  6
15  5  1  3
 7  8 10  -
```

您的第一步只能打 3 或 10 。

程式是由下面動作開始：將隨機挑選的數目對按其號碼在最終的正確順序中互換位置藉以打亂原來的順序；例如，如果所挑選的頭兩對數目正好是 7 、 12 和 2 、 14 ，則盤面的順序將如下所示：

```
 1 14  3  4
 5  6 12  8
 9 10 11  7
13  2 15  -
```

7 與 12 互換位置，2 與 14 互換位置。這個“洗牌”動作一直持續到所有磁磚完全打亂為止。本遊戲只有在洗牌期間的互換次數為偶數時才有解（這是可以證明的），因此上述例子是有解的，因為互換的次數為兩次。如果互換次數為 42 ，196 或 2538 （偶數），本遊戲都有解，但如果互換次數為 33 、 175 、或 1 （奇數）就解不開了。您可能會很高興聽到本程式的洗牌副程式永遠做偶數次數的

互換；它已經夠難了，何必弄得根本解不出來呢！

Fifteen

```
10   REM ******************
20   REM *                *
30   REM *     FIFTEEN    *
40   REM *                *
50   REM *       BY       *
60   REM *                *
70   REM *  STUART ANSTIS *
80   REM *                *
90   REM ******************
100  REM
110  REM
120  DIM N(20)
130  TEXT : HOME : VTAB 10
140  PRINT "WANT INSTRUCTIONS (Y/N) ?"
150  GET Y$
160  IF Y$ = "Y" THEN  GOSUB 930
170  TEXT : HOME
180  GOSUB 360
190  FOR J = 0 TO 15
200 H = 12 + 20 * (J / 4 -  INT (J / 4)): REM  J MOD4
210 V = 6 + 4 *  INT (J / 4)
220  HTAB H: VTAB V
230  GOSUB 700: REM  PRINT J
240  NEXT J
250  VTAB 24: HTAB 1: PRINT "TO EXIT FROM GAME, HIT 0 (ZERO)";
```

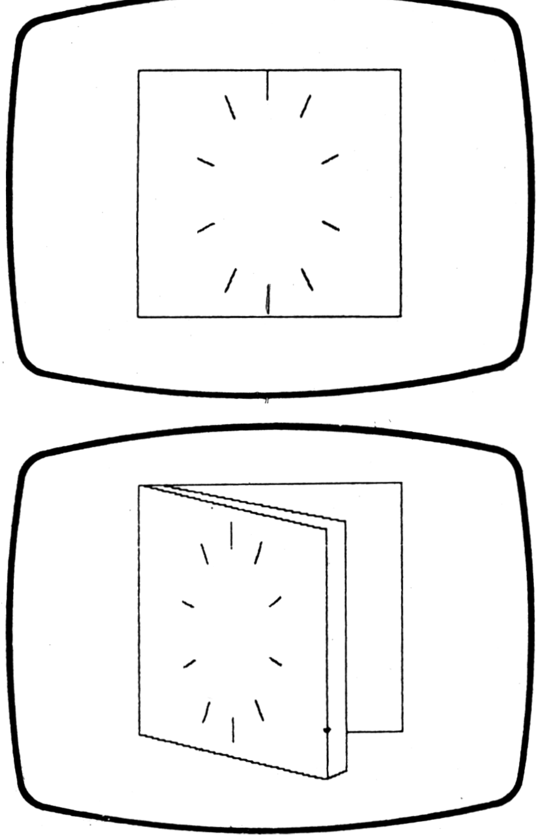
待續…

```
260  REM  MAIN LOOP-----
270  VTAB 21: HTAB 1: PRINT "TYPE YOUR NUMBER & HIT RETURN";
280  INPUT G
290  IF G = 0 THEN  GOTO 810: REM  EXIT
300  GOTO 500: REM   CHECK LEGALITY
310  VTAB 21: HTAB 31: PRINT "     ";
320  VTAB 22: HTAB 1: PRINT "
330  VTAB 23: HTAB 1: PRINT "
340  GOTO 190
350  END
360  REM  SHUFFLE-----
370  FOR J = 0 TO 15
380 N(J) = J + 1
390 N(15) = 0
400  NEXT J
410 KMAX = 10 + 2 * (30 *  RND (1))
420  FOR K = 1 TO KMAX
430 J1 = 15 *  RND (2)
440 J2 = 15 *  RND (4)
450 TT = N(J1)
460 N(J1) = N(J2)
470 N(J2) = TT
480  NEXT K
490  RETURN
500  REM  CHECK LEGALITY-----
510  FOR J = 0 TO 15
520  IF N(J) = G THEN A = J
530  IF N(J) = 0 THEN Z = J
540  NEXT J
550 AM4 = 4 * (A / 4 -  INT (A / 4)): REM   A MOD 4
560  IF Z = A - 4 AND A > 3 THEN  GOTO 610: REM  UP
570  IF Z = A + 4 AND A < 12 THEN  GOTO 610: REM  DOWN
580  IF Z = A - 1 AND AM4 <  > 0 THEN  GOTO 610: REM  LEFT
590  IF Z = A + 1 AND AM4 <  > 3 THEN  GOTO 610: REM  RIGHT
600  CALL  - 1059: VTAB 22: HTAB 35: PRINT "   ";: GOTO 260: REM  ILLEGAL

610 N(Z) = N(A)
620 N(A) = 0
630  GOTO 310
640  REM  ILLEGAL MOVE-----
650  CALL  - 1059
660  VTAB 21: HTAB 1
670  PRINT "SORRY, ILLEGAL MOVE:TRY AGAIN.
680  GOTO 260
690  END
700  REM   PRINT J-----
710  IF N(J) <  > 0 THEN  INVERSE
720  HTAB H - 1: VTAB V - 1
730  PRINT "    ";
740  HTAB H - 1: VTAB V
750  IF N(J) = 0 THEN  PRINT "    ";: GOTO 770
760  PRINT " ";N(J);: IF N(J) < 10 THEN  PRINT '
770  HTAB H - 1: VTAB V + 1
780  PRINT "    ";
790  NORMAL
800  RETURN
810  REM  END-----
820  HOME : VTAB 10
830  PRINT "WANT ANOTHER GAME (Y/N) ?"
840  GET Y$
850  IF Y$ = "Y" THEN  GOTO 170
860  VTAB 15
870  PRINT "TO EXIT, HIT 'ESC'"
880  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
890  GET H$
900  IF H$ =  CHR$ (27) THEN  HOME : VTAB 5: PRINT "OK. THANKS FOR PLAYIN
G.": END
910  PRINT : PRINT  CHR$ (4);"RUN HELLO"
920  END
```

```
930  REM  INSTRUCTIONS-----
940  HOME : VTAB 2: HTAB 16
950  PRINT "FIFTEEN"
960  PRINT : FOR M = 1 TO 8: PRINT "*****";: NEXT M
970  PRINT
980  PRINT "THIS PUZZLE IS PLAYED WITH FITEEN TILES ON A SIXTEEN-SPACE BO
     ARD."
990  PRINT "YOUR TASK IS TO RE-ARRANGE THE TILES"
1000  PRINT "INTO NUMERICAL ORDER, LIKE THIS:--"
1010  PRINT
1020  INVERSE
1030  HTAB 15: PRINT " 1  2  3  4"
1040  HTAB 15: PRINT " 5  6  7  8"
1050  HTAB 15: PRINT " 9 10 11 12"
1060  HTAB 15: PRINT "13 14 15   "
1070  NORMAL
1080  PRINT : PRINT "ON EACH MOVE, SLIDE A TILE INTO AN"
1090  PRINT "ADJACENT EMPTY SPACE BY TYPING THE"
1100  PRINT "TILE'S NUMBER AND HITTING RETURN"
1110  PRINT : PRINT
1120  INVERSE
1130  PRINT "  HIT ANY KEY TO CONTINUE...   ";
1140  GET H$
1150  RETURN
```

開鎖(Safebreaker)



顯示保險箱門開合的範例圖形

這個單人遊戲是從Creative Computing Press 出版的More Basic Computer Games 獲得靈感的，並特別爲Apple 改寫爲可利用遊戲控制器和高解析度圖形的遊戲。

假設您是一位宵小，面對著一座帶有一個四一號碼組合鎖的保險箱，您並不知道它的號碼是多少，但是知道第一和第三個號碼爲負，而第二與第四個號碼爲正；這表示您必須交互反時鐘和順時鐘方向轉動保險箱的旋鈕（0號控制器）

。當您旋轉控制器時您將會看到有個指標在保險箱上移動，而且當時旋鈕的位置也會顯示在螢光幕上一個小窗框內，您的任務就是找出那四個號碼順序。當您接近號碼時您會聽到鎖內制拴發出的一個咔喇聲，而當您找到正確號碼時，您會聽到一聲蜂鳴聲，這就表示您該繼續找下一個號碼。同時，您會聽到一個時鐘嘀嗒嘀嗒地響，而且您有可能隨時被發現…您的任務是儘快打開該保險箱；一旦您做到了，您將會看到保險箱的門旋然而開。多試幾次本遊戲，讓開鎖的時間漸漸縮短；當您熟練後邀一些朋友來玩，看看誰比較合適當罪犯。

Safebreaker

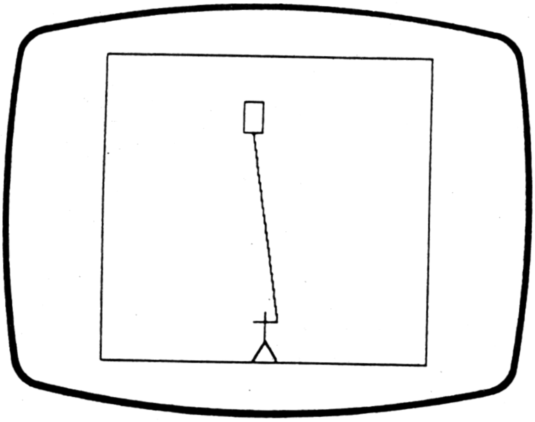
```
10   REM *******************
20   REM *                 *
30   REM *   SAFEBREAKER   *
40   REM *                 *
50   REM *       BY        *
60   REM *                 *
70   REM * STUART ANSTIS   *
80   REM *                 *
90   REM *******************
100   REM
105   TEXT : HOME : VTAB 10: PRINT "WANT INSTRUCTIONS (Y/N) ?"
110   GET Y$
115   IF Y$ = "Y" THEN  GOSUB 1200
120   TEXT : HOME : HGR2 : HGR
130 R = 55
140 F = 1:G = 1: REM  SCALE X & Y
150 K$ = "-+"
160 PI = 3.14159265
170 XC = 140:YC = 80
180   GOSUB 450
190   GOSUB 490
200   POKE 230,64: REM  HGR2
210   GOSUB 860
```

```
220  POKE 230,32: REM  HGR
230 XC = 140:YC = 80
240 TIME = 0
250  GOSUB 600
260  GOSUB 990: REM  COLLECT FOUR NUMBERS
270 J = 1: REM  1ST OF 4 NUMBERS
280 P =  PDL (0)
290 A = 2 * PI * P / 360 + 1
300 X = XC + R *  -  SIN (A)
310 Y = YC + R *  COS (A)
320  REM  DRAW ROTATING POINTER-----
330  FOR HC = 3 TO 0 STEP  - 3
340  HCOLOR= HC
350  IF HC = 0 AND  ABS (P -  PDL (0)) < 1 THEN TIME = TIME + 0.1: VTAB 2
     1: HTAB 32: PRINT "TIME="; INT (TIME);: GOTO 350
360  HPLOT XC,YC TO X,Y
370 PP = P - 123
380  VTAB 22: HTAB 19: PRINT "      ";
390  VTAB 22: HTAB 20: PRINT PP
400  NEXT HC
410 TIME = TIME + 1
420  VTAB 21: HTAB 32: PRINT "TIME="; INT (TIME);
430  GOSUB 1050: REM  CHECK POINTER POSITION
440  GOTO 280
450  REM   DRAW DOOR FRAME-----
460  HCOLOR= 3
470  HPLOT 0,0 TO 279,0 TO 279,159 TO 0,159 TO 0,0
480  RETURN
490  REM  DRAW SPOKES-----
500  HCOLOR= 3
510 R1 = 60:R2 = 78
520  FOR A = 0 TO 6.3 STEP 0.63
530 X1 = XC + R1 *  SIN (A) * F
540 X2 = XC + R2 *  SIN (A) * F
550 Y1 = YC + R1 *  COS (A) * G
560 Y2 = YC + R2 *  COS (A) * G
570  HPLOT X1,Y1 TO X2,Y2
580  NEXT A
590  RETURN
600  REM  ASTERISKS-----
610  FOR V = 21 TO 23
620  VTAB V: HTAB 18
630  PRINT "*******";
640  NEXT V
650  RETURN
660  REM  OPEN THE DOOR---
670  POKE  - 16304,0: POKE  - 16299,0: POKE  - 16297,0: POKE  - 16302,0
680  FOR Q = 1 TO 3
690  FOR T = 1 TO 300: NEXT T
700  POKE  - 16300,0
710  FOR T = 1 TO 300: NEXT T
720  POKE  - 16299,0
730  NEXT Q
740  CALL  - 1059: CALL  - 1059
750  TEXT : HOME : VTAB 10
760  PRINT "YOU OPENED THE SAFE IN "; INT (TIME);" SECONDS !!"
770  VTAB 15
780  PRINT "WANT ANOTHER TRY (Y/N) ?";
790  GET Y$
800  IF Y$ = "Y" THEN  POKE  - 16304,0: POKE  - 16300,0: POKE  - 16297,0:
      POKE  - 16301,0: GOTO 230
810  PRINT : PRINT : PRINT "HIT 'ESC' TO EXIT"
820  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
830  GET H$
840  IF H$ =  CHR$ (27) THEN  HOME : VTAB 10: PRINT "OK.  THANKS FOR PLAY
     ING.": END
850  PRINT : PRINT  CHR$ (4);"RUN HELLO"
860  REM  OPEN DOOR-----
870  HCOLOR= 3
880  GOSUB 450
890  HCOLOR= 0
900  HPLOT 220,159 TO 0,159
910  HCOLOR= 3
920  HPLOT 0,0 TO 200,30 TO 200,189 TO 0,159 TO 0,0
930  HPLOT 0,0 TO 20,0 TO 220,25 TO 220,184 TO 200,189
940 F = 0.7:G = 0.9
950 XC = 100:YC = 95
960  GOSUB 490
970 F = 1:G = 1
980  RETURN
```

待續

```
990  REM  COLLECT FOUR NUMBERS-----
1000  FOR J = 1 TO 4
1010 N(J) =  INT ( RND (2) * 120)
1020  IF J = 1 OR J = 3 THEN N(J) =  - N(J)
1030  NEXT J
1040  RETURN
1050  REM  CHECK POINTER POSITION-----
1060  IF  ABS (PP - N(J)) < 10 THEN SOUND =  PEEK ( - 16336)
1070  IF  ABS (PP - N(J)) < 1 THEN  FOR M = 1 TO 20:SOUND =  PEEK ( - 163
36): NEXT M:J = J + 1
1080 K = 1: IF J = 2 OR J = 4 THEN K = 2
1090  VTAB 24: HTAB 12: PRINT "FIND NUMBER #";J;" ("; MID$ (K$,K,1);")";
1100  IF J > 4 THEN  GOTO 660
1110  RETURN
1200  REM  INSTRUCTIONS-----
1210  HOME
1220  HTAB 15: PRINT "SAFEBREAKER"
1230  PRINT
1240  FOR M = 1 TO 8: PRINT "*****";: NEXT M
1250  PRINT : PRINT "USE PADDLE # 0 TO CRACK THE SAFE."
1260  PRINT "THE LOCK ON THE SAFE HAS A FOUR-NUMBER"
1270  PRINT "COMBINATION CODE."
1275  PRINT
1280  PRINT "TURN THE PADDLE:--"
1290  PRINT
1300  PRINT "<-- COUNTERCLOCKWISE FOR NUMBER # 1"
1310  PRINT "--> CLOCKWISE FOR NUMBER #2"
1320  PRINT "<-- COUNTERCLOCKWISE FOR NUMBER # 3"
1330  PRINT "--> CLOCKWISE FOR NUMBER #4"
1340  PRINT
1350  PRINT "YOU WILL HEAR A 'CLICK' WHEN YOU ARE"
1360  PRINT "GETTING NEAR THE NUMBER, AND"
1370  PRINT "A 'BUZZ' WHEN YOU HAVE FOUND IT."
1380  PRINT "SEE HOW FAST YOU CAN CRACK THE SAFE."
1390  PRINT : PRINT "  *** GOOD LUCK !!  ***"
1400  PRINT : INVERSE : PRINT " HIT ANY KEY TO CONTINUE ";: NORMAL
1410  GET K$
1420  RETURN
```

耍把戲者(Juggler)



保持平衡並非易事

這是一個很有趣的模擬指尖頂著一根撞球桿的程式；有一個以高解析度描繪的小矮人正在平衡一根尖端頂著一塊重量的長球桿。這球桿不會保持平衡很久，如果它倒向左邊，您必須反時鐘旋轉1號控制器將小矮人向左移以保持平衡；如果球桿向右倒，您必須順時鐘旋轉1號控制器加以補償。您的目標當然是儘可能使球桿保持平衡，愈久愈好。如果（當）它摔倒地面或牆上時，螢光幕會告訴您保持了多長時間，然後開始下一回合。

250 那一列以要求您將1號控制器擺在中央位置而展開序幕。注意：除非您照做了，否則遊戲是不會開始的；如果您坐在那兒不動，電腦也會按兵不動，但是一旦您將控制器置於中央，就會有事情發生了。360 至410 那幾列為程式的心臟，它包括了使球桿在380 那一列倒向一邊的“地心引力”方程式（X1 和 X2 為球桿頂端和底端的水平位置）。這個方程式包含一個阻尼係數DAMP，使球桿傾斜得慢一點，好像在水中一樣，讓遊戲者有較多時間去控制它；阻尼係數愈高

就愈容易平衡。當您最後捧落球桿時程式會在520以後那幾列印出結果；同時，720和730兩列會改變阻尼係數，供下一回合遊戲。如果您的表現愈來愈好，程式會藉降低阻尼係數而使下一回合難一些；相反地，如果您的表現不太好，它會設定一個較高，較容易的阻尼係數來鼓勵您。這是藉比較您最新的成績WIN與前面幾次的平均值AV來了解您的進步情形，這種小技巧可使遊戲的難易程度符合遊戲者的技術，同時維持大致相等的難易程度使遊戲對熟練者與生疏者有相同的挑戰性。

繪圖副程式置於程式開頭的20至80那幾列中，以確保得到快速描繪速度，並使畫面不致閃爍。

Juggler

```
10   GOTO 200
20   REM  PLOTTING SUBROUTINE---
30   HPLOT XB - 20,YB TO XB,YB: REM ARMS
40   HPLOT XB - 20,YB + 20 TO XB - 10,YB + 10 TO XB,YB + 20: REM LEGS
50   HPLOT XB - 10,YB + 10 TO XB - 10,YB - 5: REM BODY
60   HPLOT XB,YB TO XT,YT: REM POLE
70   HPLOT XT - W,YT TO XT + W,YT TO XT + W,YT - H TO XT - W,YT - H TO XT -
     W,YT
80   RETURN
90   REM  *******************
100   REM *                 *
110   REM *     JUGGLER     *
120   REM *                 *
130   REM *       BY        *
140   REM *                 *
150   REM *  STUART ANSTIS  *
160   REM *                 *
170   REM *******************
180   REM
190   REM
200   DIM WIN(30)
210 A1 = 100:A2 = 100:B1 = 100:B2 = 100: REM DUMMY VALUES
```

待續

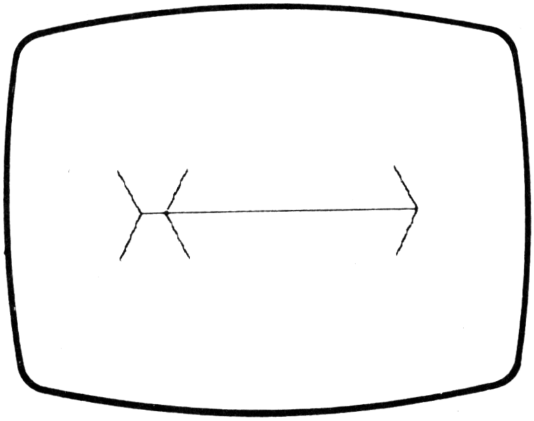
```
220 DAMP = 300: REM  INITIAL SETTING
230 W = 8:H = 16
240  TEXT : HOME : VTAB 10
250  PRINT "PLEASE TURN PADDLE # 1 TO ITS CENTRE": PRINT "POSITION TO STA
     RT THE GAME"
260  PRINT : PRINT
270  IF  PDL (1) > 120 AND  PDL (1) < 130 THEN CLICK =  PEEK ( - 16336) -
      PEEK ( - 16336): PRINT "THANK YOU.": FOR T = 0 TO 500: NEXT T: GOTO
     290
280  GOTO 270
290  HOME
300  VTAB 23: HTAB 1: PRINT "DAMPING = ";DAMP
310  VTAB 22: HTAB 1: PRINT "SCORE = ";: HTAB 14: PRINT "GAME #";
320 Y2 = 138: REM  POSITION OF FEET
330 A = 0:R = 100: REM POLE'S ANGLE & LENGTH
340  HGR : HCOLOR= 3
350  HPLOT 0,0 TO 279,0 TO 279,159 TO 0,159 TO 0,0
360 X1 =  PDL (1) + 3 * ( RND (2) -  RND (8)): REM  INITIAL PERTURBATION
370 X2 =  PDL (1) + 20
380 A = A +  ATN ((X1 - X2) / DAMP): REM   ANGLE OF POLE
390 WIN = WIN + 1
400 X1 = X2 + R *  SIN (A)
410 Y1 = Y2 - R *  COS (A)
420  IF Y1 > 155 OR X1 > 279 - W OR X1 < W THEN  GOTO 520: REM YOU DROPPE
     D IT
430 XT = A1:XB = A2:YT = B1:YB = B2
440  HCOLOR= 0: GOSUB 20
450 XT = X1:XB = X2:YT = Y1:YB = Y2
460  HCOLOR= 3: GOSUB 20
490 A1 = X1:A2 = X2:B1 = Y1:B2 = Y2: REM OLD VALUES STORED FOR LATER ERAS
     URE
500  VTAB 22: HTAB 9: PRINT WIN;: HTAB 21: PRINT NGAME + 1
510  GOTO 370
520  REM YOU DROPPED IT---
530 NGAME = NGAME + 1
540 WIN(NGAME) = WIN
550  FOR J = 1 TO NGAME
560 SUM = SUM + WIN(J)
570  NEXT J
580 AV = SUM / NGAME
590  CALL  - 1059
600  TEXT : HOME : VTAB 5
610  PRINT "YOU HIT THE ";
620  IF Y1 > 155 THEN  PRINT "FLOOR !": GOTO 640
630  PRINT "WALL !"
640  PRINT : PRINT
650  PRINT "ON GAME # ";NGAME
660  PRINT : PRINT
670  PRINT "YOU SCORED ";WIN
680  PRINT : PRINT
690  PRINT "YOUR AVERAGE SCORE SO FAR IS ";  INT (AV)
700  PRINT : PRINT
710 DAMP = 10 *  INT (DAMP * .12): REM EASIER
720  IF WIN > AV THEN DAMP = 10 *  INT (DAMP / 14.4): REM  HARDER
730 A = 0:WIN = 0:SUM = 0:AV = 0
740  GOTO 250
```



箭(Arrow)

您若畫出兩條等長的線，並在第一條線上畫上一些朝外的箭頭，在第二條線畫上一些朝內的箭頭，則第一條線會看起來比第二條線長一些，這就是Muller-Lyer 錯覺。心理學家貢獻了很多心力在這種光學錯覺的研究上，希望能從它們

了解大腦是如何處理這些視覺現象。到現在，我們仍不太清楚這些錯覺現象，這顯示了不是這門科學太艱深了，就是科學家們不夠聰明。



電腦產生的箭頭可顯示Muller-Lyer 錯覺

本程式可供您實驗箭鰭角度對Muller-Lyer 錯覺的影響。電腦爲箭鰭選擇一個角度，並以高解析度畫出兩支尾對尾的箭，您要利用遊戲控制器調整到兩支箭大致上等長，使中央的箭頭看起來好像平分該水平線一樣；當您滿意時就按下控制器旋鈕，電腦會嗶一聲，然後擦去它們，再以不同箭鰭角度畫兩支箭。您要按照上面的步驟照樣再做一次，如此繼續二十次之後，電腦會清除螢光幕，然後將您的結果統統畫出來。沿著底線（X軸）是鰭與箭身所夾的角度，在左邊，尾鰭指向左方，與箭身成10度夾角，在右邊，它們指向右方，也與箭身成10度夾角，而在原點（X軸的一半處），尾鰭與箭身成垂直。垂直的Y軸可顯示主觀錯覺的程度。如果您沒有錯覺，每次調整箭長時都正確無誤，則所有的點都應該在水平線上。實際上，大多數人的結果會看起來像有條線向上斜45度，這顯示尾鰭愈靠近箭身錯覺也就愈大。

關於光學錯覺的原因人們已獲得一點共同的看法，R. L. Gregory 相信大腦

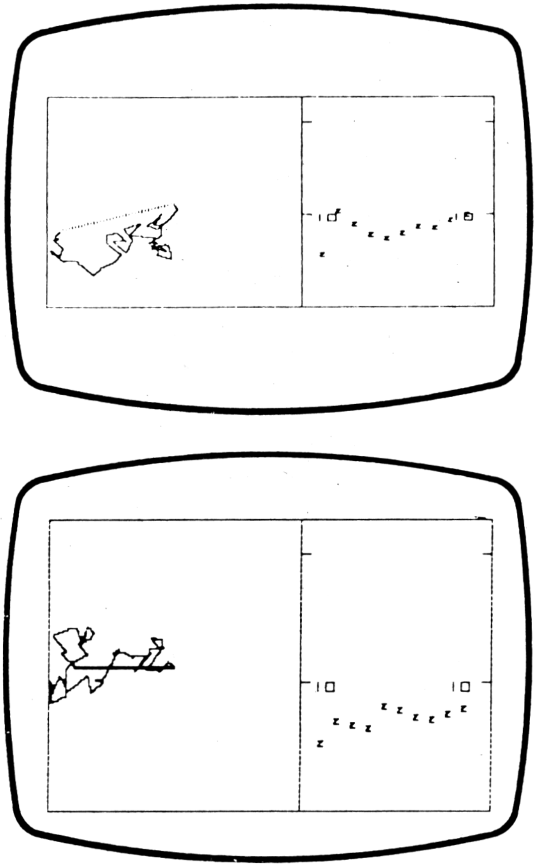
是把箭看成一幅三維空間形狀的圖像，大腦把指向外面的鰭看成好像一間房間內遠離的角落，而把指向裡面的鰭看成好像一本半開的書背，這書背向觀察者，大腦依據錯誤判斷的距離來修正所感覺的尺寸，因而導致錯覺。

Arrow

```
10  REM  *********************
20  REM            ARROW 9
30  REM  *********************
40  DIM A(100),Q(100)
50  DIM X(100),Y(100)
60 N = 1
62  TEXT : HOME : VTAB 10
64  PRINT "WANT INSTRUCTIONS (Y/N) ?  "
66  GET Y$
68  IF Y$ = "Y" THEN  GOSUB 1000
70  HGR2
80 NN = 3
90  PRINT
100  HCOLOR= 7
110  REM  MAIN LOOP -----
120 A(N) = 20 / 57.3 *  INT (9 *  RND (1) - 4)
130 A = A(N)
140 C = 20 *  RND (1) - 20 *  RND (1): REM  RANDOM OFFSET ---
150 R =  PDL (1)
160 Q = R / 2 - 60 + C: REM  S'S SETTINGS ---
170 X1 = 40:X2 = 240:Y = 100:RR = 25
180 L =  ABS (X1 - X2): REM  STALK LENGTH
190 Z = (X1 + X2) / 2
200 RC = RR *  COS (A):RS = RR *  SIN (A)
210  GOSUB 320
220  IF  ABS (R -  PDL (1)) > 2 THEN F = 1: HCOLOR= 0: GOSUB 340: HCOLOR=
     7: GOSUB 280: REM  RE-PLOT ARROW IF SUBJECT ALTERS PADDLE ---
230 X =  PEEK ( - 16286)
```

```
240  IF F = 1 AND X > 127 THEN  GOSUB 410: GOTO 110
250  IF N > 20 THEN  GOTO 480: REM  END EXPERIMENT AFTER 20 TRIALS ---
260  IF X < 128 THEN  GOTO 220
270  GOTO 230
280  REM
290 R =  PDL (1)
300 Q = R / 2 - 60: REM  S'S SETTINGS ---
310 Q = Q + C: REM  RANDOM OFFSET ---
320  REM  PLOT ARROWS -----
330  HPLOT X1,Y TO X2,Y: REM   STALK ---
340  HPLOT X1 - RS,Y + RC TO X1,Y TO X1 - RS,Y - RC: REM  LEFT FIN ---
350  HPLOT X2 - RS,Y + RC TO X2,Y TO X2 - RS,Y - RC: REM  RIGHT FIN ---
360  HPLOT Z + Q + RS,Y + RC TO Z + Q,Y TO Z + Q + RS,Y - RC: REM  CENTRE
      FIN ---
370  RETURN
410  REM   ERASE LAST PICTURE -----
420 Q(N) = Q
430  CALL  - 1059: REM  BEEP
440  HCOLOR= 0: GOSUB 340
450  HCOLOR= 7
460 N = N + 1
470  RETURN
480  HGR : REM  PLOT GRAPH OF RESULTS -----
490  HCOLOR= 7
500  HPLOT 140,0 TO 140,160
510  HPLOT 60,80 TO 220,80
520  FOR K = 0 TO 160 STEP 20
530 XK = K + 60:YK = K
540  HPLOT XK,78 TO XK,82
550  HPLOT 138,YK TO 142,YK
560  NEXT K
570  HCOLOR= 5
580  FOR J = 1 TO N
590 X(J) = A(J) * 57.3 + 140
600 Y(J) = 4 * Q(J) + 80
610  IF Y(J) < 0 THEN Y(J) = 0
620  IF Y(J) > 159 THEN Y(J) = 159
630  HPLOT X(J),Y(J) TO X(J) + 1,Y(J) TO X(J) + 1,Y(J) + 1 TO X(J),Y(J) +
     1 TO X(J),Y(J)
640  NEXT J
650  PRINT "X=FIN ANGLE (0=PERPENDICULAR)"
660  PRINT "Y=ILLUSION (0=NO ILLUSION)"
670  PRINT "HIT SPACE BAR FOR TABLE OF RESULTS"
680 X =  PEEK ( - 16384)
690  IF X > 127 THEN  TEXT : GOTO 710
700  GOTO 680
710  POKE  - 16368,0: REM  RESET KEYBOARD STROBE
720  REM  PRINT TABLE OF RESULTS -----
730  PRINT "TRIAL #","ANGLE","SETTING"
740  FOR J = 1 TO N - 1
750  PRINT J,A(J) * 57.3, INT (Q(J) * 10) / 10
760  NEXT J
770  PRINT "HIT SPACE BAR TO SWITCH BETWEEN GRAPH": PRINT "AND TABLE OF R
     ESULTS.  FOR ANOTHER": PRINT "RUN, PRESS BUTTON ON SUBJECT'S PADDLE.
     "
780  TEXT
790 X =  PEEK ( - 16384)
800  IF X > 127 AND  PEEK ( - 16297) <  > 0 THEN  POKE  - 16297,0: POKE  -
     16304,0: POKE  - 16368,0: GOTO 820: REM  CHANGE OVER TO GRAPH
810  IF X > 127 AND  PEEK ( - 16297) = 0 THEN  POKE  - 16303,0: POKE  - 1
     6298,0: POKE  - 16368,0: REM   CHANGE OVER TO TABLE
820  IF  PEEK ( - 16286) > 127 THEN  CALL  - 1059: GOTO 60
830  GOTO 790
1000  REM  INSTRUCTIONS -----
1010  HOME : VTAB 5
1020  GOSUB 1500
1030  PRINT
1040  HTAB 8
1050  PRINT "MULLER-LYER ARROW ILLUSION"
1060  PRINT
1070  GOSUB 1500
1080  PRINT : PRINT : PRINT "SET LENGTH OF ARROWS BY TURNING PADDLE"
1090  PRINT "# 1 UNTIL BOTH ARROWS LOOK THE SAME"
1095  PRINT "LENGTH.  THEN PRESS PADDLE BUTTON."
1098  PRINT
1100  PRINT "DO THIS TWENTY TIMES."
1110  PRINT : PRINT : PRINT
1120  INVERSE
1130  PRINT "HIT ANY KEY TO CONTINUE...   "
1135  NORMAL
1140  GET K$
1150  RETURN
1500  REM  ASTERISKS -----
1510  FOR J = 1 TO 40: PRINT "*";: NEXT J
1520  RETURN
```

東倒西歪(Random Walk)



"東倒西歪 "的兩個範例

亂數（ random number ）在數學與物理中是極具重要性的，您可以利用 Apple 內建的亂數產生器（ random generator ）來研究它們的特性（參閱第一章中"亂數"那一節）。

這個高解析度程式可展示二維空間東倒西歪的特性。假想有一位醉漢由廣場中央的一支燈柱開始以一米寬的步幅在散步，但是他實在醉得太厲害了，以致無法穩定地朝一個方向走；相反地，他每一步的方向都是隨機的。問題是：當他走

了 10 步，或 100 步，或 1000 步，…，他倒底能走多遠？很明顯地，答案不可能是一個固定的量，但必定有一個平均的數字；研究結果顯示他從燈柱所走的平均距離約等於N的平方根，此處的N爲他所走的步數，這完全是根據概率來計算的。我們試考慮他的前兩步：如果他的運氣特別好，則這兩步將會在同一方向上，因而他從起點所走的總距離將爲 2×N。如果他的運氣特別差，則他的第一步向外跨出之後第二步又會將他帶回原來位置；換句話說，他的第二步爲第一步的 180 度，他從起點所走的距離爲 0 。實際上，平均來說，他的運氣不會特別好，也不會特別差，第二步多半與第一步垂直，因而他從燈柱所走的距離此刻爲兩邊各爲 1 米長的直角三角形的斜邊。根據畢達格拉斯（ Pythagoras ） 定理（ 畢氏定理），這個距離爲 2 的平方根。同理可證N步之後的距離爲N的平方根。

本程式可在高解析度螢光幕上並排顯示出兩幅圖：左邊一幅模擬醉漢的足跡（總共有 50 步），並在最後從起點至終點畫出一條直線；這條直線的長度會顯示在螢光幕底部，並且畫在螢光幕右邊的圖上。程式會執行 20 次，每次都會把總距離顯示出來。右圖所畫的不是最近一次所走的距離，而是截至當時各次路線的平均距離。您將可看到所涵蓋的距離每次雖然都不相同，但平均起來大約在 7 左右（約等於 50 的平方根）。

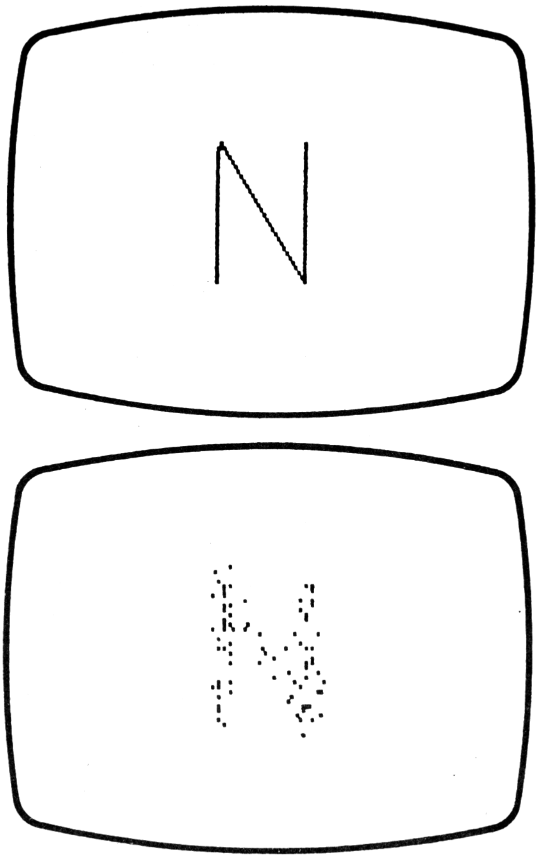
Random Walk

```
10   REM *******************
20   REM *                 *
30   REM *   RANDOM WALK   *
40   REM *                 *
50   REM *       BY        *
60   REM *                 *
70   REM *  STUART ANSTIS  *
80   REM *                 *
90   REM *******************
100  REM
110  REM
195  TEXT : HOME : VTAB 10
200  REM : M=WALK#. N=STEP#. NMAX=# OF STEPS PER WALK. S= TWICE THE AVERA
     GE STEP LENGTH.
210  REM : D= DISTANCE FROM START AT END OF LAST WALK. DBAR=D AVERAGED OV
     ER ALL WALKS SO FAR. Y(I) HOLDS VALUES OF DBAR.
220  REM : SC=SCALING FACTOR IN PLOTTING GRAPHS.
230  REM : MMAX = NUMBER OF WALKS
250  REM   ***********************
300 S = 10:SC = 7:NMAX = 50
305 MMAX = 10
310  DIM Y(100)
312  FOR I = 1 TO 100:Y(I) = 0: NEXT I
315  PRINT "WANT INSTRUCTIONS (Y/N)?"
318  GET Y$
320  IF Y$ = "Y" THEN  GOSUB 2900: REM  FOR EXPLANATORY TEXT ************
     ***
325  HOME : VTAB 21
328  HGR
330  FOR M = 1 TO MMAX
350  HCOLOR= 7
360  REM : PLOT FRAME ******************
370  HPLOT 0,0 TO 159,0 TO 159,159 TO 0,159 TO 0,0
375  HPLOT 159,0 TO 279,0 TO 279,159 TO 159,159
380  REM : PLOT SCALE MARKS *************
390  FOR L = 0 TO 2
400  HPLOT 159,159 - SC * S * L TO 165,159 - SC * S * L
405  HPLOT 273,159 - SC * S * L TO 279,159 - SC * S * L
410  NEXT L
415  REM  PLOT DIGITS OF "10" *****
420 YY = 159 - SC * S
425 WW = 5
430  FOR XX = 170 TO 255 STEP 85
440  HPLOT XX,YY TO XX,YY + WW
450  HPLOT XX + WW,YY TO XX + WW,YY + WW TO XX + 2 * WW,YY + WW TO XX + 2
      * WW,YY TO XX + WW,YY
460  NEXT XX
500  REM : PLOT RANDOM WALK **************
510 ASTART = 80:BSTART = 80
515 A = ASTART:B = BSTART
520  HPLOT ASTART,BSTART
530  FOR N = 1 TO NMAX
540 K = 1: REM  J,K=+1 OR -1, DETERMINING DIRECTION OF RANDOM STEPS *****
    **
550 Z =  RND (1)
560  IF Z > 0.5 THEN K =  - 1
570 A = A + S * K *  RND (2)
580 J = 1
590 Z =  RND (2)
600  IF Z > 0.5 THEN J =  - 1
610 B = B + S * J *  RND (3)
620 AP = A:BP = B
630  IF A < 0 THEN AP = 0
635  IF A > 279 THEN AP = 279
640  IF B < 0 THEN BP = 0
645  IF B > 159 THEN BP = 159
650  HPLOT  TO AP,BP
660  NEXT N
700  REM : CALCULATE DISTANCE FROM START ***************
710 H =  ABS (A - ASTART)
720 V =  ABS (B - BSTART)
730 D =  INT (100 * 2 * ( SQR (H ^ 2 + V ^ 2)) / S) / 100
740  PRINT "WALK # ";M;"=";D;" STEPS FROM START
750 T = T + D
760 DBAR =  INT (100 * T / M) / 100
765 Y(M) = DBAR
770  PRINT "AVERAGE STEPS=";DBAR
780  HCOLOR= 2
790  HPLOT ASTART,BSTART TO AP,BP
800  HPLOT ASTART + 1,BSTART TO AP + 1,BP
810  HPLOT ASTART,BSTART + 1 TO AP,BP + 1
```

待續

```
850  REM : PLOT GRAPH ON RIGHT ***********
860  FOR I = 1 TO M
870  HCOLOR= 3
880 P = 160 + 10 * I:Q = 160 - SC * Y(I)
890  HPLOT P,Q TO P + 3,Q TO P,Q + 3 TO P + 3,Q + 3
900  NEXT I
990  REM : ERASE PREVIOUS WALK *****
1000  HCOLOR= 0
1010  FOR V = 1 TO 158
1020  HPLOT 1,V TO 158,V
1030  NEXT V
1990  NEXT M
2000  END
2900  FOR I = 1 TO 20: PRINT : NEXT I
3000  PRINT "A DRUNKARD, OR A MOLECULE, TAKES A WALK OF ";NMAX;" RANDOM-L
     ENGTH STEPS IN RANDOM": PRINT "DIRECTIONS.  ";
3010  PRINT "THIS PROGRAM SHOWS THAT THEAVERAGE DISTANCE FROM THE STARTIN
     G POINT ";
3020  PRINT "WILL BE EQUAL TO THE SQUARE ROOT OF ";NMAX;" (ABOUT "; INT (
     10 *  SQR (NMAX)) / 10;")"
3025  PRINT
3026  PRINT "THERE WILL BE ";MMAX;" WALKS ALTOGETHER."
3028  PRINT
3030  PRINT "YOU WILL SEE THE RANDOM WALKS ON THE"
3032  PRINT "LEFT.  THE AVERAGE DISTANCE, OVER ALL"
3034  PRINT "WALKS SO FAR, WILL BE PLOTTED ON"
3036  PRINT "RIGHT."
3040  FOR I = 1 TO 8: PRINT : NEXT I
3050  INVERSE : PRINT "  HIT ANY KEY TO CONTINUE   ": NORMAL
3060  GET H$
3080  RETURN
9999  END
```

擴散(Scatter)



本圖顯示一幅開始時與經過 30 秒亂漂後的情況

這是另外一個"東倒西歪"（ random walk ）程式，我們要從一個以高解析度圖形描繪在螢光幕的字母N開始。這個字母由一組點組成，而每個點以每步1微米以及各自獨立的方向亂漂。每個點持續地進行，結果使得該字母逐漸地擴散，好像糖在熱水中溶化一樣（未經攪拌）。開始時，該字母好像很快地分解開來，但您會很驚訝它居然要花那麼長的時間才能使它散得無法識別；這完全是因爲平方根定理的緣故。如果漂移5微米平均需要N秒，則漂移5公分（前者之十

倍）平均所需的時間並非 10 ×N 秒，而是可能百倍之多。

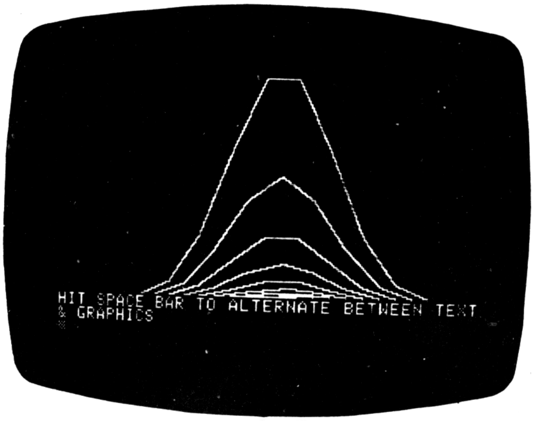
Scatter

```
100  REM    **********************
110  REM
120  REM          SCATTER
130  REM             BY
140  REM        STUART ANSTIS
150  REM
160  REM   ************************
170  HGR
180  DIM XO(90),YO(90),XN(90),YN(90)
190  REM :DRAW INITIAL FIGURE ***********
200  FOR I = 1 TO 30
210  HCOLOR= 7
220 XO(I) = 125
230 YO(I) = I + 80
240  HPLOT XO(I),YO(I)
250  NEXT I
260  FOR I = 31 TO 60
270 XO(I) = I + 95
280 YO(I) = I + 50
290  HPLOT XO(I),YO(I)
300  NEXT I
310  FOR I = 61 TO 90
320 XO(I) = 155
330 YO(I) = I + 20
340  HPLOT XO(I),YO(I)
350  NEXT I
360  HOME
370  VTAB 22: PRINT "TO RETURN TO MENU AT ANY TIME, HIT": PRINT "ANY KEY"
```

待續…

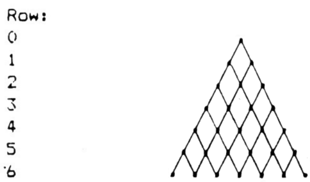
```
380  REM : INCREMENT THE SCATTER AND REPLOT THE FIGURE ***************
390  FOR I = 1 TO 90
400  HCOLOR= 7
410 Z =  RND (1)
420 A = 1
430  IF Z > 0.5 THEN A =  - 1
440 PUSH =  PEEK ( - 16384)
450  IF PUSH > 127 THEN 590
460 XN(I) = XO(I) + A
470 B = 1
480 Z =  RND (4)
490  IF Z > 0.5 THEN B =  - 1
500 YN(I) = YO(I) + B
510  HPLOT XN(I),YN(I)
520  REM : ERASE OLD POSITION OF FIGURE ****************
530  HCOLOR= 0
540  HPLOT XO(I),YO(I)
550 XO(I) = XN(I)
560 YO(I) = YN(I)
570  NEXT I
580  GOTO 390
590  REM  KEY PRESSED---
600  POKE 16368,0: REM  CLEAR KEYBOARD
610  TEXT : HOME : VTAB 10
620  PRINT "OK THEN. BACK TO THE MENU."
630  FOR T = 1 TO 1000: NEXT T
640  PRINT : PRINT  CHR$ (4);"RUN HELLO"
650  END
```

巴斯卡三角形(Pascal's Triangle)



巴斯卡三角形的圖形顯示

假設下面是一組以鑽石形狀配置的街道圖：



從上面來的人每走到一個叉路口都會猶豫到底要走左邊或右邊的路。若從第 0 列開始，向左走會帶他們到第 1 排的東邊那一點，向右走則會帶他們到西邊那一點，我們可用 1, 1 表示到達那兩點的走法。現在，第 2 排有 3 點，要到中間那一點的走法有兩種：您可以取消前面所選的向左走而改爲向右走，或者前面先選右邊的人可以改走左邊，而到西邊那一點或東邊那一點的走法只有一種，我們可用 1 ， 2 ， 1 來表示到達這些點的可能走法。我們若從 1 ， 2 ， 1 將相鄰的數目加起來可得到 1 ， 3 ， 3 ， 1 ，這也就是第 3 排各點的可能走法。用這種方法繼

續往下推，我們就可得到〝巴斯卡〞三角形：

```
Row:                                        Total:
0                       1                       1
1                     1   1                     2
2                   1   2   1                   4
3                 1   3   3   1                 8
4               1   4   6   4   1              16
5             1   5  10  10   5   1            32
6           1   6  15  20  15   6   1          64
7         1   7  21  35  35  21   7   1       128
8       1   8  28  56  70  56  28   8   1     256
```

在每一列中，第一個數目與最後一個數目都是1，而其他數目則分別是前一排中該數目頂上兩個數目的和。圖中最後一列的數目分別代表人們可以到達第八列的第一，第二，…，第九點的方法，而到達這一列的各種可能走法的總數爲$1+8+28+56+70+56+28+8+1=256=2^8$。

人的高度可以看成是許多因素在生長過程中作用的總和，其中有些是促進生長的，有些是妨碍生長的。我們可以用擲錢幣來模擬這些因素的效果，看看是在自然生長過程中額外加一吋呢，還是從現有身高減掉一吋。如果您這樣做八次，而且如果有256個人也同樣這樣做，我們可以根據上面的行人過街模式來想像其可能之結果；我們可以把失去一吋身高看成向左走，而把增加一吋看成向右走。我們最後會發現身高的分佈就像巴斯卡三角形的最後一列一樣，比平均身高少四吋的人只有一位，比平均身高少三吋的人有8位，少兩吋的人有 28 位，…，高三吋的有8位，高四吋的只有1位。如果我們將所有身高相等的人排成一列，我們可以有九列，而且如果令這九列與牆壁垂直，則具有平均身高的人所形成的那一列將會最長。如果從空中俯看，各列的長度將會形成一個〝鐘形〞曲線（bell curve）或常態曲線（normal curve）。

這個小程式可以印出巴斯卡三角形中的數目，並且可以將它們描繪成圖形。

120 那一列將沿著三角形兩邊的數目都設定為1，而130 那一列求取其餘所有數目的總和，這些數目是在140 那一列中PRINT出來，並在250 至300 那幾列中HPLOT。190 至240 那幾列可供使用者隨便敲任何一個鍵就可來回選擇印出文字（TEXT）或高解析度圖形。GET 敍述檢查是否有鍵被按下，而190 和220 那兩列中的POKE 將TEXT 頁和高解析度圖形的第1 頁置於相交互換的位置。這些被稱為〝圖形軟體開關〞（"graphics soft switches"的POKE指令可參閱Applesoft 參考手册第132～133 頁的詳細說明。如果您希望巴斯卡三角形超過 15 列，則您應該刪去200 那一列，否則有些數目會重疊在一起。

Pascal's Triangle

```
10  REM PASCAL'S TRIANGLE---
20  TEXT : HOME
30  PRINT "PASCAL'S TRIANGLE"
40  PRINT
50  INPUT "HOW MANY ROWS ";RMAX
60  DIM N(RMAX,RMAX)
70  REM CALCULATE SUMS-----
80  HGR : HCOLOR= 3: TEXT : HOME
90 N(0,0) = 1
100  FOR R = 1 TO RMAX
110  FOR C = 1 TO R
120 N(R,1) = 1:N(R,R) = 1
130 N(R,C) = N(R - 1,C) + N(R - 1,C - 1)
140  PRINT N(R,C); SPC( 1);
150  IF R < 11 THEN  GOSUB 250
160  NEXT C
170  PRINT
180  NEXT R
190  POKE  - 16304,0: POKE  - 16297,0: REM POKE-16300,0
200  VTAB 21: PRINT "HIT SPACE BAR TO ALTERNATE BETWEEN TEXT & GRAPHICS"
210  GET G$
220  POKE  - 16303,0: REM DISPLAY TEXT
230  GET G$
240  GOTO 190
250  REM  HIRES PLOT---
260  POKE 230,32: REM PLOT TO PAGE 1
270 X = 140 + 20 * C - 10 * R:Y = 159 - N(R,C)
280  IF C = 1 THEN  HPLOT X,Y
290  HPLOT  TO X,Y
300  RETURN
```

魔術方塊(Magic Squares)



魔術方塊在數學上已有很長久的歷史，
我們現在要把其產生過程自動化。

一個“N階魔術方塊”是指一個具有N×N元素的陣列，其每一列，每一行，甚至對角線的數目加起來都要等於N×（N^2+1）/2。

下面是一個3 × 3 的魔術方塊：

```
4 9 2
3 5 7
8 1 6
```

每一列，每一行，和對角線加起來都等於 15 。下面是一個四階魔術方塊，其每一列，每一行和對角線加起來都等於34：

```
16  3  2 13
 5 10 11  8
 9  6  7  2
 4 15 14  1
```

魔術方塊在古時候的中國就已經發現了，相傳有一天禹帝（公元 2000 年前）站在黃河邊時曾看到一隻神龜背上就有這種魔術方塊。Albrecht Durer 遠在

1514 年也曾在他有名的Melancholia 畫中包含了一個上面所示的四階魔術方塊，中間靠下的兩列數目記載的是他作畫的日期。福蘭克林曾導出一個產生魔術方塊的簡單方法，Ball 和Coxeter 在他們合著的Mathematical Recreations and Essays 書中也說明了數種方法。最近，York 大學（位於美國的Toronto）的Ian Howard 發明了一個又新且多用途的方法，它不僅可以產生任何階數的魔術方塊，而且可以推廣至三維空間，或者任何空間維數的N維空間超級魔術方塊！本節所提供的程式是筆者與Howard 在1980 年五月份的Creative Computing 所發表的，其方法是根據廣泛用於統計學和休閒數學的拉丁方塊（Latin Square）。N階拉丁方塊是一個包含N×N 元素，其最前面N個整數會出現N次，但在每一列，每一行，和對角線中都出現一次，而且只能出現一次。下面是兩個不同的5×5拉丁方塊範例：（習慣上我們使用0至4，不是1至5；如果您喜歡1至5，只要把拉丁方塊中的數目全部加一。）

3	4	0	1	2
1	2	3	4	0
4	0	1	2	3
2	3	4	0	1
0	1	2	3	4

4	1	3	0	2
3	0	2	4	1
2	4	1	3	0
1	3	0	2	4
0	2	4	1	3

因爲0，1，2，3，4 在每一列，每一行和對角線上都有，因此每一列，每一行和對角線加起來都爲0＋1＋2＋3＋4 ＝10。

現在，讓我們將這兩個拉丁方塊重疊在一起成爲一個新方塊，這個新方塊的每一個元素各由一個拉丁方塊內取出一個數字所組成：

34	41	03	10	22
13	20	32	44	01
42	04	11	23	30
21	33	40	02	14
00	12	24	31	43

這種組合稱爲〝希臘一拉丁式方塊〞(Greco-Latin square) ,或歐氏方塊(Eulerian square) 以紀念瑞士數學家 Leonhard Euler (1707 — 1783) 。每一個二位數數目在這種方塊中只出現一次,而且任何一列,一行或對角線上數目的左邊數字加起來都等於 0 +1 +2 +3 +4 = 10;右邊的數字也是一樣。現在——這是一個關鍵步驟——將每一對數字當作以 5 爲底(五進位) 的數目;以 5 爲底的數目從 00 至 44 等於十進位的 0 至 24 ,它們均只出現一次。此外,因爲每一列,每一行或對角線上右手邊或〝個位數〞的和等於一個常數,左手邊或〝五位數〞也是一樣,因此我們得到的是一個五進位數目的魔術方塊。要驗證這一點,我們可以將他們轉換爲熟悉的十進位數目(將個位數+ 5 ×左手邊數字);例如,34 (以 5 爲底)= 3 × 5 + 4 × 1 =19 (以10爲底)。因此,希臘拉丁式方塊可轉換爲如下的十進位形式:

```
19 21  3  5 12 = 60
 8 10 17 24  1 = 60
22  4  7 13 15 = 60
11 18 20  2  9 = 60
 0  7 14 16 23 = 60
==============
60 60 60 60 60
```

每一列,每一行,和對角線的和都是 60 。

本程式的做法是先產生兩個N × N的拉丁方塊,再將它們加以重疊來產生成對的數字,把這些數字對當作兩位數的N進位(以N爲底) 數目,再將它們轉換爲十進位形式。N 必須是一個値數,但可以愛多大就多大——如果您願意,它可以超過百萬;不過,如果N太大,您可能需要等一段時間才能看到結果!如果您想要知道一些有關魔術方塊的事情,可讀W. S. Andrews 著的 Magic Squares and Cubes (Dover 書局,1960);如果您不想,您不必擔心數學問題,因爲

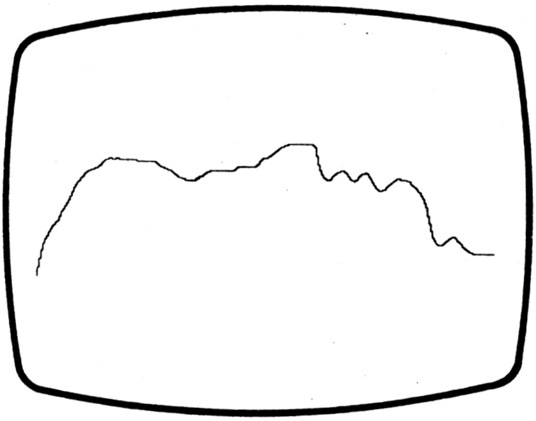
您根本不必了解數學就可以RUN本程式。

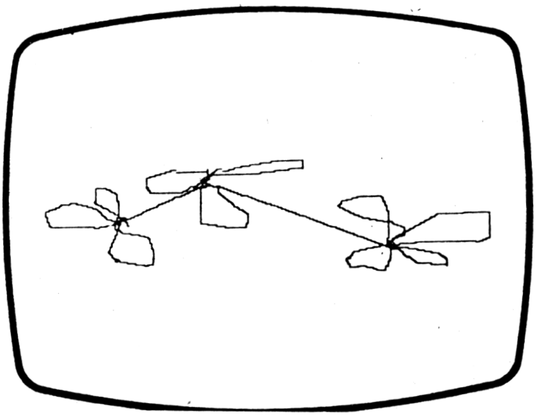
Magic Squares

```
10  REM  ************************
20  REM
30  REM  MAGIC SQUARES AND CUBES
40  REM
50  REM    BY STUART ANSTIS
60  REM
70  REM  ************************
72  REM
91  TEXT : HOME : VTAB 10
92  PRINT "DO YOU WANT A 2-D MAGIC SQUARE OR A 3-D MAGIC CUBE?"
94  INPUT "TYPE 2 FOR A SQUARE, 3 FOR A CUBE.....";D$
95 D = VAL (D$)
96  IF D < > 2 AND D < > 3 THEN GOTO 94
98  IF D > 2 THEN GOTO 116
100  PRINT "WHAT SIZE MAGIC SQUARE? ";
105  PRINT "TYPE A PRIME"
110  INPUT "NUMBER GREATER THAN 3.....";N
112  IF 2 * (N / 2 - INT (N / 2)) < 0.1 THEN GOTO 110
114  GOTO 125
116  PRINT "WHAT SIZE MAGIC CUBE? ";
118  INPUT "TYPE A PRIME NUMBER EQUAL TO OR GREATER THAN 11.....";N
120  IF 2 * (N / 2 - INT (N / 2)) < 0.1 THEN GOTO 118
125 J = 0
128  PRINT : PRINT : PRINT
135  DIM V(N)
140  DIM S(N,N),T(N,N),U(N,N)
145  DIM ROW(N),COL(N)
150  DIM J(N,N)
160  REM  ************************
170  REM
180  REM        FIRST ROW
190  REM  INSERT DIGITS FROM 1 TO
200  REM   N IN RANDOM ORDER
210  REM
220  REM  ************************
230  PRINT "ONE MOMENT, PLEASE."
240  PRINT
300  FOR C = 1 TO N
310 V(C) = INT (N * RND (1))
320  FOR K = 1 TO C - 1
330  IF V(C) = V(C - K) THEN GOTO 310
340  NEXT K
350 U(1,C) = V(C)
360  NEXT C
400  REM  ********************
410  REM     SHIFT DIGITS 2 PLACES
420  REM  TO RIGHT IN SUCCESSIVE
430  REM           ROWS
440  REM      L=(C-2) MOD N
450  REM  ********************
500  FOR R = 2 TO N
510  FOR C = 1 TO N
520 L = N * ((C - 3) / N - INT ((C - 3) / N)) + 1
530 U(R,C) = U(R - 1,L)
540  NEXT C
550  NEXT R
```

```
600  REM  ********************
610  REM  SHIFT DIGITS 4 PLACES
620  REM  TO RIGHT IN SUCCESSIVE
630  REM        PLANES
635  REM  J=(C-4*P) MOD N
640  REM  ********************
700  FOR P = 1 TO N ^ (D - 2)
710  FOR R = 1 TO N
720  FOR C = 1 TO N
730 J = N * ((C - 4 * P) / N -  INT ((C - 4 * P) / N)) + 1
732 J =  INT (J + 0.5)
740 S(R,C) =  INT (U(R,J))
745  NEXT C: NEXT R
750  REM  ********************
760  REM  U(R,C) FLIPPED LEFT/RIGHT
770  REM  AND UP/DOWN;
780  REM  T(R,C)=N*(L/R MIRROR IMAGE) +
790  REM        N*N*(U/D MIRROR IMAGE)
795  REM  ********************
797  FOR R = 1 TO N: FOR C = 1 TO N
800 T(R,C) = S(R,C) + N * S(R,N - C + 1) + N * N * S(N - R + 1,C) * (D -
  2)
810 SS =  INT (T(R,C)) + 1
820 ROW(R) = ROW(R) + SS
830 COL(C) = COL(C) + SS
840  REM  FORMATTING STATEMENT *****
850 T = 5 -  LOG (SS + 1) /  LOG (10)
855  REM :PRINT THE MAGIC SQUARE *****
860  PRINT  SPC( T);SS;
865 T(R,C) = 0
866 SS = 0
870  NEXT C
880  REM  PRINT SUMS OF ROWS *****
900  PRINT "  ***  ";ROW(R)
910 ROW(R) = 0
920  PRINT
930  NEXT R
1000  REM : PRINT SUMS OF COLUMNS *****
1010  FOR C = 1 TO N
1020  PRINT "*****";
1030  NEXT C
1040  PRINT
1050  FOR C = 1 TO N
1060 T = 5 -  LOG (COL(C)) /  LOG (10)
1070  PRINT  SPC( T);COL(C);
1080  NEXT C
1090  FOR C = 1 TO N:COL(C) = 0: NEXT C
1100  PRINT : PRINT : PRINT
1200  NEXT P
1210  PRINT "WANT ANOTHER ONE? (Y/N)  "
1220  GET Y$
1225  PRINT : PRINT
1230  IF Y$ = "Y" THEN  GOTO 160
1240  PRINT "TO EXIT, HIS 'ESC'"
1250  PRINT "HIT ANY OTHER KEY TO RETURN TO MENU"
1260  GET H$
1270  IF H$ =  CHR$ (27) THEN  HOME : VTAB 5: PRINT "OK. THANKS FOR PLAYI
  NG.": END
1280  PRINT : PRINT  CHR$ (4);"RUN HELLO"
2000  END
```

動畫描圖板(Animated Sketchpad)





本程式可提供您一維（上圖）或二維（下圖）
空間的高解析度繪圖能力

本程式其實應算爲繪圖程式或單人遊戲，也不知爲什麼，筆者稱它爲科學程式，將它放在本章中。它可模擬三種功能使電腦成爲一個在下列三領域中非常有用的工具：(1)電子學，(2)動態畫影片製作，與(3)工廠。

(1)電子工程師常常利用信號產生器（ function generator ）產 生某種形式隨時間變化的電壓波形，並不斷地重複。這個工作以往常用類比方法（ analog techniques ）來執行 ，特別是當所需波形爲一單純正弦波，斜線波（ ramp ）

，或方波的時候。但是，現在已漸漸被微處理機取代了，尤其當所需波形不規則時。波形是藉指定波形中許多點的瞬間電壓值用數位方式產生的；產生器通常在陣列中儲存 256 ， 512 或 1024 個數值，然後經由數位一至一類比轉換器（digital-to-analog converter ）很快地連續輸出，產生隨時間變化的電壓波形。

⑵電腦已漸漸被用來產生電影中所需的動態特殊效果；華德廸斯奈的電影 Tron 就是一個很好的例子。電腦可以很容易地產生各種形狀，但很多人還不了解它也可以很容易地控制運動的路徑。卡通藝術家可以用手畫一條軌跡，讓一支唱針在一座圓形唱盤上移動，而電腦可以〝看著〞並記下他手的移動，然後藉卡通動態描繪技巧沿著完全相同的路徑重畫出來。這種方法可將圖形好壞立即回饋給藝術家，讓他可以反覆修改，直到滿意爲止。

⑶電腦可成爲很好的機器人。在工廠裡，電腦可以設置來〝看著〞一位裝配員執行某些熟練的工作（諸如在車床上車一具燭台），將該裝配員所做的各個步驟記下來，然後照著做一次，十次，甚或一年，讓該裝配員沒有事做（這對裝配員是很糟的事！）。電腦可看作一台特製的錄音機，但不是記錄音樂，而是記錄一連串三維空間的動作；這些動作包括打開車床的電源，鎖緊夾具，進刀等動作。如果由人來做這些動作，可稱爲技術，但如果由電腦來做這些事，它們只是〝錄放音〞而已——錄放一組在裝配員敎導機器時在〝錄音〞階段輸入陣列中的數字座標，然後再在機器接管時在〝放音〞階段讀出那些數值。

請注意，這三項功能——波形產生，電影動態效果，以及機器人模仿動作——執行的過程實際上是相同的，只是它們分別是在一維、二維和三維空間中進行。

〝動畫描圖板〞是藉將遊戲控制器或搖桿的動作儲存在一個X，Y座標的陣

列中而模擬這種過程的。該陣列是在20 那一列中藉DIM 宣告的，而使用者要在50和60那兩列中決定要一維空間，抑或二維空間。一維空間可供您記錄並重放一個由左至右橫跨螢光幕的高解析度波狀線，而二維空間可讓您利用搖桿描繪任何二維空間的高解析度圖案。要在程式中加入第三維Z來讀取第三個控制器並不是很難的事。三個可變電阻器的軸要彼此垂直，它們可以放在一支特製機械臂的肩部，肘部或腕部。這三個可變電阻器（標定爲0號，1號和2號控制器）可供程式在機械臂描過某個複雜三維空間物體時接受三維空間的輸入值。

80至180那幾列在螢光幕上畫出一個高解析度點。在一維空間裡，該點由螢光幕的左邊開始，使用者藉旋轉1號控制器來調整其垂直位置(Y)，按下1號控制器按鈕就會把這個Y值存入陣列中（190至310那幾列），然後將該點向右移動1微米（280那一列）。使用者再調整第二點的高度，並按下按鈕，如此依序從左至右在螢光幕上建立一條波狀線。當該點到達螢光幕的右手邊線時（仍然是280那一列），控制權交到330至410那幾列，然後會有一小段垂直線（在390那一列畫出）從左至右橫過螢光幕並上下跳動，就好像它的軌跡與剛剛指定的波狀線完全一樣似的！Y值是從陣列Y(J)中讀出來的，如果有一片數位一至一類比轉換器連接至輸出端，這些Y值便會將您的Apple變成一台多用途的任意波形產生器。

在二維空間裡，80至180那幾列就像Etch-a-Sketch*程式（參閱第一章“多利用遊戲控制器”一節）一樣，將該點的位置置於搖桿的控制之下（比用兩只控制器要好得多）。但該點不會留下永久的軌跡，除非你按下1號控制器按鈕（150那一列），它才會在螢光幕上畫出一個白點，並將X和Y值存入陣列中（230，240那兩列）。在該點移動時連續按按鈕可在螢光幕上描繪出一條連續的

白線，但是，按 0 號按鈕會把“筆”提離紙張（ 140 那一列），可以令該點移到別的位置而不會留下軌跡；再按 0 號按鈕可將筆再度放下繼續描繪。

220 那一列中的 PAUSE（暫停）與打字機鍵盤上的自動重複（ auto-repeat ）功能的作用相同；每次短促地按一下 1 號按鈕會儲存一個 X，Y 位置，但是如果連續按著按鈕，該點的位置便會從搖桿連續地被讀取，直到按鈕鬆掉爲止。這表示我們可以利用搖桿沿著任何希望的路徑，這裡快一些，那裡慢一點，把一條隨速度變化的長軌跡輸入電腦。當軌跡輸入完成之後，使用者可以把搖桿置於左下角，亦即將 0 號與 1 號控制器都設定爲零，然後按 1 號按鈕從“記錄”模式進入“重放”（ playback ）模式就可以產生動態電影了！有一條垂直線（ 390 那一列）會沿著您剛描繪的二維空間路徑移動，並反映出您描繪圖形時的速度。

您可以當做一個練習，將本程式納入您自己的電視遊戲中，利用它來控制。譬如說一個平面或一艘太空船的變距（ Zooming ）與攫取畫面。利用搖桿來變距或攫取鏡頭，將 X 和 Y 以資料檔（ data file ）儲存在磁碟片上，要玩時再令遊戲將該檔案讀出磁碟片。如果您想進入三維空間的眞實世界，爲什麼不買或製作一支三維機械臂以及三個類比－至－數位轉換器？這樣您就可以一面躺在椅子上，一面讓 Apple 整天製作東西，看着財源滾滾而來。

Animated Sketchpad

```
10   REM  ANIMATED SKETCHPAD---
20   DIM X(300),Y(300)
30   TEXT : HOME
40   PRINT "HOW MANY DIMENSIONS: 1 OR 2"
50   GET D
60 S = 2: REM STEP SIZE WHEN D=1
70   HGR
80 X =  INT ( PDL (1) * 278 / 255) * (D = 2)
90 Y =  INT (154 -  PDL (0) * 154 / 255)
100   REM MAKE POINTER VISIBLE---
110   FOR HC = 3 TO 0 STEP  - 3
120   HCOLOR= HC
130   HPLOT X,Y
140   IF  PEEK ( - 16287) > 127 THEN PENWRITE = 0: REM   PUSHBUTTON #0 LIF
      TS PEN
```

待續

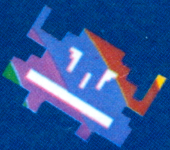
```
150  IF HC = 3 AND  PEEK ( - 16286) > 127 THEN  GOSUB 190: REM  PUSHBUTTO
     N #1 READS X, Y INTO ARRAY
160  NEXT HC
170  IF D = 2 THEN  GOTO 80
180  GOTO 90
190  REM COLLECT X(J), Y(J)---
200  REM  TO EXIT, SET BOTH PADDLES TO ZERO AND PRESS PUSHBUTTON #1
210  IF  PDL (0) = 0 AND  PDL (1) = 0 THEN 330: REM  EXIT
220  FOR T = 0 TO 30: NEXT T: REM PAUSE=AUTO-REPEAT KEY
230  IF D = 2 THEN X(J) = X: IF J = 1 THEN  HPLOT X,Y TO X + 1,Y + 1
240 Y(J) = Y
250  IF PENWRITE = 0 THEN PENWRITE = 1: GOTO 270
260  IF J > 1 THEN  HPLOT XOLD,YOLD TO X,Y
270 XOLD = X:YOLD = Y
280  IF D = 1 THEN X = S * J: IF X > 260 THEN 330
290  VTAB 21: PRINT "X=";X;"  Y=";Y;"  "
300 J = J + 1
310  RETURN
320  REM PLOT MOTION PATH---
330  HGR : HOME
340 JMAX = J - 1
350  FOR J = 1 TO JMAX
360  FOR HC = 3 TO 0 STEP  - 3: HCOLOR= HC
370 X = S * J
380  IF D = 2 THEN X = X(J)
390  HPLOT X,Y(J) TO X,Y(J) + 5
400  NEXT HC,J
410  GOTO 350
```

版 權 所 有
翻 印 必 究

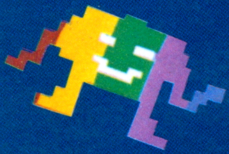
自己設計Apple遊戲

原著發行日期：1983 年
原著書名：WRITE YOUR OWN APPLE GAMES
原 著 者：Stuart Anstis
譯　　者：郭　念　台
發 行 人：楊　鏡　秋
出 版 者：儒 林 圖 書 有 限 公 司
地　　址：台 北 市 重 慶 南 路 一 段 111 號
電　　話：3812302　3110883　3140111
郵政劃撥：0106792-1號
吉 豐 印 刷 廠 有 限 公 司 承 印
板橋市三民路二段正隆巷 46 弄 7 號
行 政 院 新 聞 局 局 版 台 業 字 第 1492 號

中華民國七十三年十二月 初版
定價新台幣 180 元正



自己設計 APPLE 遊戲



包含的遊戲超過 50 種，除了可供您在Apple 上自娛外，本書還有詳細的說明，告訴您如何用BASIC 自己動手設計遊戲！

程式包括：

動作遊戲：

坦克大決戰，衝刺，轟炸機，伺機而發，賽馬，等等。

策略遊戲：

浪子回頭，搭橋，股票市場，黃金大拍賣，等等。

文字遊戲：

劊子手，拼字遊戲，生日卡，貴姓大名，等。

繪圖遊戲：

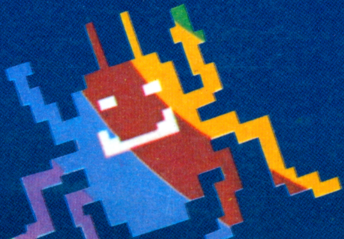
蝴蝶，烟火，李沙育圖，蒲公英，等。

教育應用：

箭，東倒西歪，擴散，魔術方塊，等。

玩玩每個遊戲，再研究研究第一章介紹的手法，副程式和思維方法，您就可以動手設計自己的Apple 遊戲了！

儒林圖書公司 印行



自己設計 APPLE 遊戲

郭念台 譯

儒林
SIM-102